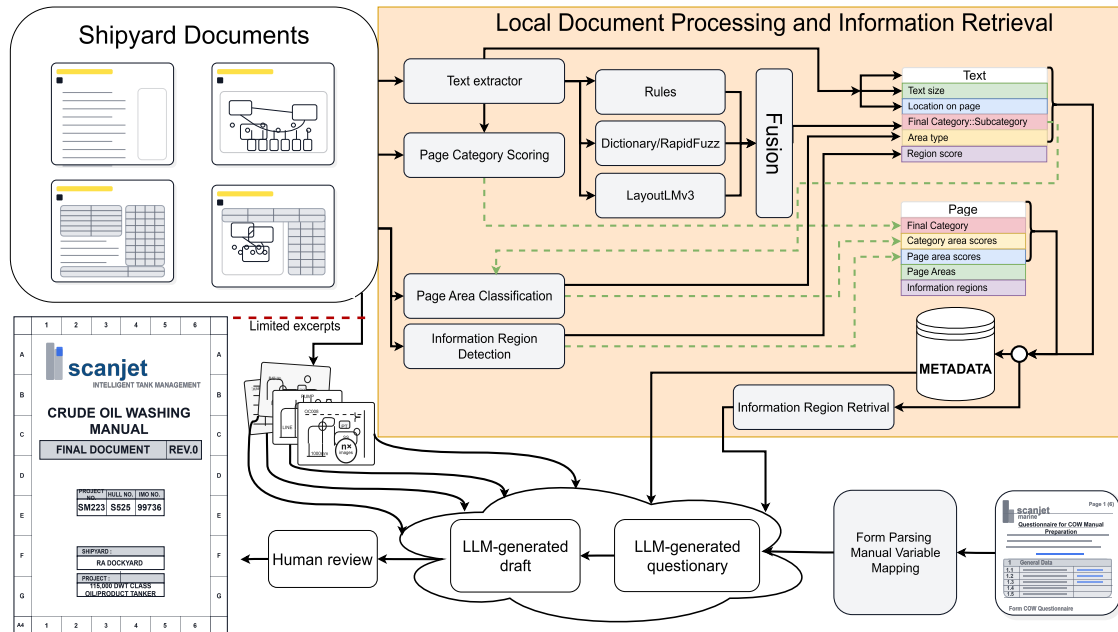




Use of AI techniques to extract data from CAD drawings

To create a COW (Crude Oil Washing) manual

Master's thesis in Mobility Engineering

RASMUS ARVIDSSON

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2026

www.chalmers.se

MASTER'S THESIS IN MOBILITY ENGINEERING

Use of AI techniques to extract data from CAD drawings

To create a COW (Crude Oil Washing) manual

RASMUS ARVIDSSON



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Use of AI techniques to extract data from CAD drawings
To create a COW (Crude Oil Washing) manual
RASMUS ARVIDSSON

© RASMUS ARVIDSSON 2026.

Supervisor: Johan Hemlab, Scanjet
Examiner: Wengang Mao, Department of Mechanics and Maritime Sciences

Master's Thesis 2026
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: A flow chart of the data retrieval process and report generation

Typeset in L^AT_EX
Gothenburg, Sweden 2026

Use of AI techniques to extract data from CAD drawings
To create a COW (Crude Oil Washing) manual
RASMUS ARVIDSSON
Department of Mechanics and Maritime Sciences
Division of Marine Technology
Chalmers University of Technology

Abstract

Crude oil washing (COW) manuals require vessel-specific technical information from drawings, questionnaires and system documentation. Much of this information is available in documents provided by the shipyard, but it is distributed across drawing pages, tables, labels and diagram regions, making manual extraction time-consuming and sensitive to missing or inconsistent source material. This thesis asks to what extent AI-based techniques can support the production of COW manuals from CAD-derived drawings and related technical documents with varying formats, layouts and information content. Confidential source documents, complete page images, full OCR output and system-connectivity information were processed locally. Only selected metadata and bounded excerpts assessed as insufficient to reconstruct a drawing or its connection logic were made available to a cloud-based large language model (LLM).

The work implemented a multi-stage pipeline for page categorization, text extraction, token-level semantic labelling, structural page analysis, target-region detection, retrieval and LLM-assisted draft generation. The source material described in the dataset appendix contained 32 project entries from anonymized Chinese and Korean shipyards. The pipeline combined direct PDF text extraction, OCR, TF-IDF page-category scoring, rules, dictionary matching, LayoutLMv3 token classification, YOLO-based page-content and target-region detection, YAML-guided search and an evidence-controlled LLM workflow with local preprocessing.

The final page-category model classified 64 of 67 validation pages correctly, corresponding to 95.5 % accuracy. The final directed-search retrieval reached 96.5 % validation box recall, 94.4 % page recall and 66.5 % candidate precision. In a one-project proof-of-concept case study, the workflow generated a draft of the cargo oil system section and an LLM-generated questionnaire. Of the 24 questionnaire fields, 11 were supported by the available evidence and 13 remained unresolved. The draft included the system family, tank arrangement, main cargo oil pump data, major pipe dimensions and stripping-system information, while control, monitoring, heating and several operational fields were omitted or marked for review. The case study demonstrates workflow feasibility but does not estimate generalization to new projects.

The results show that AI methods can support parts of COW manual production, particularly evidence location, structuring and preparation of a traceable first draft on the evaluated material. The system is not suitable as an autonomous manual-generation workflow, and no conclusion can be drawn about performance on new projects from the available LLM case study. Human engineering review and addi-

tional structured source material remain necessary for completeness, approval quality and operational safety.

Keywords:Crude Oil Washing, Marpol Annex I, COW manual, document AI, OCR, LayoutLMv3, YOLO, retrieval, LLM, CAD drawings, maritime engineering.

Preface

This report presents a master's thesis project carried out at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology during the spring of 2026. The work was performed in collaboration with Scanjet Marine AB and concerns the use of AI-based methods to extract technical information from vessel documentation for use in crude oil washing manual preparation.

The project was shaped by two practical conditions. First, the source material consisted of real shipyard documentation with varying quality, layout and document structure. Second, the material was confidential and therefore had to be processed within a local Scanjet-controlled environment. These conditions influenced both the technical choices and the evaluation strategy. The thesis should be read as a feasibility study of an evidence-assisted workflow, rather than as a complete production system for automatic manual generation.

Acknowledgements

I would like to thank Scanjet Marine AB for providing the thesis assignment, the technical material, and the working environment needed to carry out the project. I am especially grateful to my supervisor Johan Hemlab for his guidance, domain knowledge, and feedback throughout the work. I would also like to thank André Liljegren for helping to organize the thesis project and for his continued support throughout the work, and Hardiyanti Razak for preparing the shipyard documents used in the project.

I would also like to thank my examiner, Wengang Mao, at the Department of Mechanics and Maritime Sciences, Chalmers University of Technology, for guidance and support.

Finally, I would like to thank Omar Alabdalla and Alamin Alreda, master's students in computer science who were carrying out another thesis project at Scanjet at the same time. Their discussions, suggestions, and pointers regarding AI methods were valuable during the development of this work.

Rasmus Arvidsson
Gothenburg, October 2026

List of Acronyms

Below is the list of acronyms used throughout this thesis, listed in alphabetical order:

AI	Artificial Intelligence
API	Application Programming Interface
CAD	Computer-Aided Design
CAP	Capacity Plan
COW	Crude Oil Washing
COS	Cargo Oil System
CPU	Central Processing Unit
CSV	Comma-Separated Values
DN	Nominal Diameter
DWT	Deadweight Tonnage
DWG	Drawing file format
DXF	Drawing Exchange Format
FPSO	Floating Production, Storage and Offloading unit
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GUI	Graphical User Interface
IDF	Inverse Document Frequency
IGS	Inert Gas System
IGTV	Inert Gas and Tank Ventilation Systems
IMO	International Maritime Organization
IWBS	International Work Breakdown Structure
JSONL	JSON Lines
LLM	Large Language Model
MARPOL	International Convention for the Prevention of Pollution from Ships
ML	Machine Learning
OCR	Optical Character Recognition
ODME	Oil Discharge Monitoring Equipment
PDF	Portable Document Format
P&ID	Piping and Instrumentation Diagram
PN	Nominal Pressure
PV	Pressure/Vacuum
RAG	Retrieval-Augmented Generation

RAM	Random-Access Memory
SFI	Skipsteknisk Forskningsinstitutt (Ship Research Institute of Norway)
SOLAS	International Convention for the Safety of Life at Sea
TF-IDF	Term Frequency–Inverse Document Frequency
TWS	Tank Washing System
UDP	Upper Deck Plan
ULCC	Ultra Large Crude Carrier
VLCC	Very Large Crude Carrier
WBS	Water Ballast System
YAML	YAML Ain't Markup Language
YOLO	You Only Look Once

Nomenclature

Below is the nomenclature of indices, sets, metrics and variables used throughout this thesis.

Indices and Sets

Symbol	Description
c	Class or label category
C	Number of classes in a classification problem
d	Document or project document set
e	Expert index in a fusion model
p	Page in a document
r	Detected or labelled region on a page
t	Token or extracted text string
$\mathcal{C}$	Set of page, token or region classes
$\mathcal{D}$	Set of project documents
$\mathcal{P}$	Set of pages in the processing corpus
$\mathcal{R}$	Set of labelled or detected information regions
E	Number of experts in a fusion model

Evaluation Metrics

Metric	Description
TP	True positives
FP	False positives
TN	True negatives
FN	False negatives

P	Precision, defined as $TP/(TP + FP)$
R	Recall, defined as $TP/(TP + FN)$
$F1$	Harmonic mean of precision and recall, defined as $2PR/(P + R)$
IoU	Intersection over Union between predicted and reference bounding boxes
$mAP50$	Mean average precision at an IoU threshold of 0.50
$mAP50-95$	Mean average precision averaged over IoU thresholds from 0.50 to 0.95

Model and Retrieval Variables

Variable	Description
$\hat{y}$	Predicted class label
y	Reference class label
$\mathbf{x}$	Input feature vector for a token, page or region
$\mathbf{s}$	Score vector returned by a model or rule branch
$\mathbf{s}^{(rule)}$	Score vector from the rules expert
$\mathbf{s}^{(dict)}$	Score vector from the dictionary expert
$\mathbf{s}^{(layout)}$	Score vector from the LayoutLMv3 expert
$\mathbf{s}^{(fusion)}$	Combined score vector from the fusion layer
$\mathbf{z}$	Fused class-score vector
λ_e	Scalar contribution weight for expert e
W	Matrix of class-specific expert weights
$\mathbf{g}(\mathbf{x})$	Input-specific gating vector
b	Bounding box coordinates for a token or detected region
q	Query or target field used during retrieval

Units and Drawing Terms

Term	Description
mTH	Metres total head, used for pump head ratings
m ³ /h	Cubic metres per hour, used for flow capacity
A	Nominal pipe-size suffix used in several shipyard drawings, for example 650A

DN	Nominal pipe diameter designation
PN	Nominal pressure designation

Contents

List of Acronyms	x
Nomenclature	xiii
List of Figures	xix
List of Tables	xxi
1 Introduction	1
1.1 Background	1
1.1.1 Crude Oil Washing (COW)	1
1.1.2 MARPOL Annex I	3
1.1.3 Crude Oil Washing Manual	3
1.1.4 Questionnaire	5
1.2 Study Material and Technical Setup	6
1.2.1 Source Documents	6
1.2.2 Computational Environment	6
1.2.3 Development Support	7
1.3 Purpose	7
1.4 Research Question	7
1.5 Limitations / Demarcations	7
1.5.1 Data confidentiality and handling restrictions	7
2 Theory	9
2.1 Document Formats and Representations	9
2.1.1 File formats	9
2.2 Engineering Documents	10
2.2.1 Standards	13
2.3 AI/ML Methods	13
2.3.1 Evaluation Metrics	14
2.3.2 Text-Based Machine Learning Approaches	15
2.3.3 Computer Vision	18
2.3.4 Multimodal Model Architectures	19
3 Methods	22
3.1 Local Document Processing and Information Retrieval	23
3.2 Text Extraction	24

3.3	Page Category Scoring	25
3.4	Word Semantic Labelling	26
3.4.1	Rules	28
3.4.2	Dictionary	29
3.4.3	LayoutLMv3	29
3.4.4	Fusion	30
3.5	Page Area Classification	30
3.6	Valuable information regions	32
3.6.1	Information Region Detection	33
3.7	Final Embeddings	34
3.8	Information Region Retrieval	35
3.9	LLM Agent	35
3.9.1	Local Evidence Access and Data Boundary	36
3.9.2	LLM-Generated Questionnaire	37
3.9.3	Manual Drafting and Review	38
3.9.4	Case-study Evaluation	39
4	Results	40
4.1	Documents, Data and Computational Environment	40
4.2	Overall Result	41
4.3	Text Extraction Results	41
4.4	Page-category Results	42
4.5	Word Semantic Labelling Results	42
4.6	Structural Page Analysis Results	42
4.7	Target-region Detection Results	43
4.8	Area Scoring and Retrieval Results	43
4.9	Pipeline Timing and Practical Performance	44
4.10	LLM-Assisted Manual Drafting Results	44
4.10.1	LLM-Generated Questionnaire Results	44
4.10.2	Manual Draft Output	45
5	Conclusion	46
5.1	Limitations	46
5.2	Future Work	47
	Bibliography	49
A	Detailed Validation Results	I
A.1	OCR Tool-selection Results	I
A.2	Page-category Validation	I
A.3	Semantic-labelling Validation	II
A.4	Structural Page Analysis	IV
A.5	Target-region Detection	V
A.6	Directed-search Class Results	V
A.7	Project-level Processing Times	VI
B	Software Inventory for the Metadata Pipeline	VII

B.1	In-House Programs Used in the Metadata Pipeline	VII
B.2	Third-Party Software used in the Metadata Pipeline	VIII
C	Plotting Drawings with Embedded Text in AutoCAD	XI
C.1	Handling Missing SHX Files	XI
C.2	PDF Plot Configuration	XI
D	Dataset	XIII

List of Figures

1.1	Scanjet SC-30T tank cleaning machine (AB, 2026a)	2
1.2	Working principle of a tank cleaning machine head (AB, 2026a) . . .	2
1.3	Example cover used to represent the vessel-specific COW manual in the overall workflow.	4
1.4	Schematic representation of a shipyard questionnaire, used to identify the questionnaire component in the overall workflow.	5
1.5	Schematic representation of shipyard document pages, used to identify the source-document input in the overall workflow.	6
2.1	Raster image enlarged to show individual pixels and their RGB values.	10
2.2	Simplified P&ID of a dual-pump arrangement, showing pumps, valves, filters and instrumentation.	11
2.3	Pump configuration in an engineering drawing.	12
2.4	Sequential representation of text units.	16
2.5	Regular grid representation of a raster image.	18
2.6	Example of an irregular graph with nodes and directed edges.	21
3.1	Overview of the workflow from shipyard documents through local document processing and controlled evidence transfer to LLM-assisted questionnaire completion, manual drafting and engineering review. . .	23
3.2	Expanded view of the local document processing and information retrieval block in the overall workflow.	24
3.3	Semantic-labelling workflow combining rules, dictionary-based fuzzy matching and LayoutLMv3 through late fusion.	26
3.4	Example training page with semantic text-label categories overlaid as coloured boxes.	31
3.5	Examples of labelled information regions: ballast-pump arrangement, cargo segregation and ballast-equipment data.	32
3.6	Page- and text-level information stored in Final Embeddings.	34
3.7	Controlled LLM-assisted drafting workflow and evidence boundary. .	36
3.8	Review table for the LLM-generated questionnaire, showing supported and unresolved fields.	38
3.9	Draft-review interface showing generated text and linked source evidence.	39
C.1	AutoCAD dialog for selecting a replacement SHX font file.	XI

C.2 AutoCAD PDF plot configuration used to preserve machine-readable text.	XII
--	-----

List of Tables

1.1	Drawings to be included in the COW manual.	5
2.1	Binary confusion matrix for one class evaluated against all other ob- servations.	14
2.2	Example df and idf values for words across Shakespeare’s 37 plays. . .	18
3.1	Page-category classes.	25
3.2	TextModel token labels.	27
3.3	Base LayoutLMv3 training configuration.	29
3.4	Training settings for the page-content YOLO detector.	31
3.5	Page-content YOLO classes.	31
3.6	AreaScore classes.	32
3.7	LabelFinal YOLO classes.	33
3.8	Training settings for the deployed <i>LabelFinal</i> YOLO detector.	33
3.9	Data boundary for the LLM-assisted workflow.	36
4.1	Dataset summary from Appendix D and the final processing split. . .	40
4.2	Hardware and software environment used for the reported analysis and computation-time measurements.	41
4.3	Overall semantic-labelling performance on the held-out test split. . .	42
4.4	Summary metrics for the page-content YOLO detector.	43
4.5	Final validation summary for actual directed-search candidate classes.	43
4.6	Processing times per page for the main document-analysis stages. . .	44
4.7	Assessment of the one-project LLM case study.	45
A.1	OCR source comparison scores.	I
A.2	Validation accuracy for the page-category model variants.	I
A.3	Page-category confusion matrix for the final mixed three-way model with sequence smoothing. Rows are true classes and columns are predicted classes.	II
A.4	Class-wise validation metrics derived from the final page-category confusion matrix.	II
A.5	Held-out test split summary for the rules expert. Categories were ranked by class-wise F1 score.	II
A.6	Held-out test split summary for the dictionary expert.	III
A.7	Held-out test split summary for the layout expert.	III

A.8	Class-wise recall, precision, and F1 for the fusion model on the held-out test split, excluding labels with support below 20.	IV
A.9	Category-wise comparison of the old page-content YOLO model and the latest merged four-class page-content YOLO model on the same 78-image validation split.	IV
A.10	Dominant page-content confusion matrix.	V
A.11	Combined raw YOLO-only performance across label classes. Recall and precision are reported using the project evaluation rule: a prediction is counted as a true positive if the ground-truth box center lies inside the prediction or if $\text{IoU} \geq 0.1$	V
A.12	Per-class validation metrics for the deployed LabelFinal YOLO detector. Classes with zero validation support are excluded.	V
A.13	Per-class validation results for the final directed-search model excluding proxy-only capacity classes.	VI
A.14	Separate proxy-recall statistics for the two capacity classes.	VI
A.15	Per-project processing time summary for the full final-embeddings run.	VI

1

Introduction

1.1 Background

1.1.1 Crude Oil Washing (COW)

Crude oil is a naturally occurring fossil fuel formed over geological time and refined into a wide range of petroleum products, including transportation fuels and petrochemical feedstocks ((EIA), 2026a, 2026b). It varies in composition depending on where it is extracted. Lighter crude oils generally contain a larger proportion of light hydrocarbons, while heavier crude oils require more extensive processing in the refinery ((EIA), 2026b).

After crude oil is extracted from the ground it needs to be transported to a refinery, except in arrangements such as FPSOs where the refinement is done on site. Sea transport in specialized crude oil carriers is one of the main solutions where sea access is available, although pipelines, trains and trucks are also used. Common large crude oil tanker classes include Aframax (80-120 k DWT), Suezmax (140-150 k DWT) and VLCC vessels (200-320 k DWT), while the largest historical ULCCs exceeded 500 000 DWT (Britannica, 2026). Following the Erika incident in December 1999, IMO accelerated the phase-out schedule for single-hull tankers, which further reduced the long-term viability of older very large tanker designs (IMO, 2026b).

During the discharge of a crude oil tanker, two problems related to residual crude oil build up that reduces the effective cargo-carrying capacity of the vessel generally occur. If the crude oil is heavy and requires heating, some of the oil sticks to the relatively cold sides of the tanks. The second is that since the oil originates from the ground, some amount of sand is also always present and will build up as sediment. These accumulated residues and sediment must be managed as part of tank cleaning and stripping operations (International Maritime Organization, 2026). One effective way to mitigate this is through crude oil washing. In this method, heated cargo is sprayed by a tank cleaning machine onto the tank surfaces in order to heat and liquefy the adhering oil and loosen sediment. Figure 1.1 shows an example of such a tank cleaning machine, while Figure 1.2 illustrates the operating principle of its rotating head. The loosened residues can then follow the cargo flow and be extracted from the tank with the cargo pump (International Maritime Organization, 2026).

This thesis was conducted in collaboration with Scanjet, a supplier of tank-cleaning equipment and vessel-specific COW documentation (AB, 2026b). Scanjet provided the historical project documents and domain context used in the study.



Figure 1.1: Scanjet SC-30T tank cleaning machine (AB, 2026a)

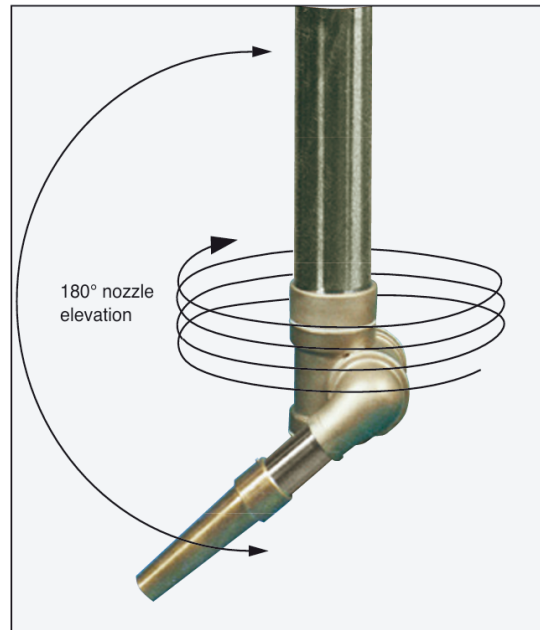


Figure 1.2: Working principle of a tank cleaning machine head (AB, 2026a)

1.1.2 MARPOL Annex I

The International Maritime Organization (IMO) is an organization within the United Nations. In 1948, an international conference in Geneva adopted a convention formally establishing IMO, and Article 1(a) of the Convention summarizes its role as providing machinery for cooperation among governments on shipping, maritime safety, navigational efficiency and the prevention and control of marine pollution from ships (IMO, 2026a).

IMO's first major task was to adopt a new version of the International Convention for the Safety of Life at Sea (SOLAS), which was achieved in 1960 (IMO, 2026a).

Although safety was and remains one of IMO's most important responsibilities, pollution quickly emerged as another major issue. The growth in the amount of oil transported by sea and in the size of oil tankers became a particular concern, and the Torrey Canyon disaster in 1967, in which 120 000 tonnes of oil were spilled, demonstrated the scale of the problem (IMO, 2026a).

When tracing the sources of oil pollution, cargo tank cleaning also emerged as a contributing operational source of marine pollution. MARPOL Annex I and the associated crude oil washing regime addressed this by reducing pollution from routine tank cleaning operations. Instead of washing cargo tanks with water, applicable tankers used crude oil as the washing medium. This reduced the generation of oily wash water, improved recovery of cargo residues, and thereby lowered the risk of operational discharges to the marine environment (IMO, 2026a; International Maritime Organization, 2026).

These regulations also created the requirement for the supplier of tank cleaning equipment to provide a crude oil washing manual specific to each vessel (International Maritime Organization, 2026).

1.1.3 Crude Oil Washing Manual

Figure 1.3 shows an example of the cover of a vessel-specific COW manual. This graphical representation is also used in the overall workflow on the thesis cover to identify the final manual output.

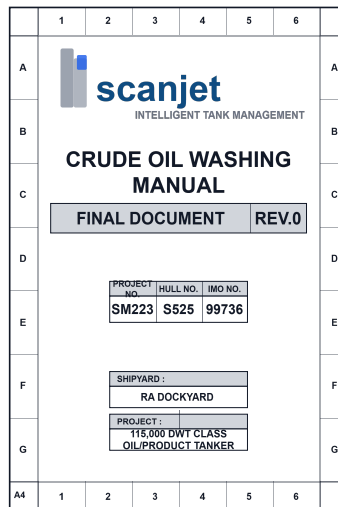


Figure 1.3: Example cover used to represent the vessel-specific COW manual in the overall workflow.

The crude oil washing manual follows a standardized IMO format and covers the operational and technical information required for the approved crude oil washing arrangement (International Maritime Organization, 2026).

This manual is developed in accordance with the requirements of MARPOL Annex I and the standard format recommended by IMO for crude oil washing operations and equipment manuals (International Maritime Organization, 2026).

The purpose of the manual is to ensure that crude oil washing is carried out in accordance with the applicable IMO requirements for the design, operation, and control of crude oil washing systems, including IMO Resolution A.446(XI) as amended by Resolutions A.497(XII) and A.897(21) (International Maritime Organization, 1979, 1981, 1999). The manual provides standard operational guidance in planning and performing a crude oil washing programme and describes safe procedures to be followed (International Maritime Organization, 2026).

A.446(XI) is the main document that governs the structure of the COW manual by specifying both its purpose and its required contents. The revised specifications define the main operational areas that must be addressed, and Section 7 lists the information to be included in the Operations and Equipment Manual, such as system description, washing procedures, stripping arrangements, ballast procedures, safety precautions, and crew responsibilities. As a result, the COW manual is structured around the requirements laid down in this IMO document rather than being freely designed by the shipyard or operator. It also specifies a set of line drawings to be included in the manual, as summarized in Table 1.1.

Table 1.1: Drawings to be included in the COW manual.

No.	Drawing content
1	Crude oil washing lines and valves
2	Cargo pumps, lines and valves
3	Ballast systems (where fitted)
4	Stripping systems
5	Position of tank washing machines
6	Position of holes for hand dipping and tank gauges
7	Inert gas deck distribution system

The manual is approved by the Administration, or on its behalf, and no alteration or revision should be made without prior approval from the approving authority (International Maritime Organization, 2026).

1.1.4 Questionnaire

To collect the data required to write the tank cleaning manuals, Scanjet requests that the shipyard building the vessel fill out a questionnaire with data about the equipment onboard, as well as, at a minimum, the drawings and system information required by the crude oil washing manual format (International Maritime Organization, 2026). Figure 1.4 shows a schematic example of this type of questionnaire. Its numbered sections and tabular response fields illustrate how vessel and equipment particulars are collected in a consistent structure before the manual is prepared. The same representation is used for the questionnaire component in the overall workflow on the thesis cover.

The image shows a schematic of a questionnaire form. At the top left is the 'scanjet marine' logo. At the top right is 'Page 1 (6)'. The title is 'Questionnaire for COW Manual Preparation'. Below the title are several horizontal lines for text input. A blue horizontal line separates the title section from the table section. The table is titled '1 General Data' and has five rows labeled 1.1, 1.2, 1.3, 1.4, and 1.5. Each row has two columns for data entry. At the bottom of the form is the text 'Form COW Questionnaire'.

Figure 1.4: Schematic representation of a shipyard questionnaire, used to identify the questionnaire component in the overall workflow.

1.2 Study Material and Technical Setup

1.2.1 Source Documents

The study material consisted of technical PDF documents from completed shipbuilding projects supplied for the thesis. Figure 1.5 provides a schematic representation of the different document pages used as input to the workflow. This representation is also used for the source-document input in the overall workflow on the thesis cover. The document sets differed considerably in size and content. Some projects contained only the drawings most directly related to the COW manual, while others also contained equipment lists, supplementary diagrams, completed COW manuals or questionnaires.

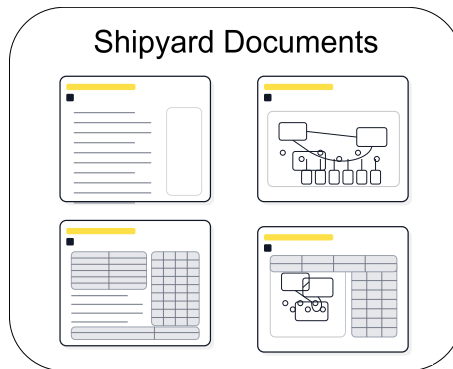


Figure 1.5: Schematic representation of shipyard document pages, used to identify the source-document input in the overall workflow.

The source projects came from anonymized Chinese and Korean shipyards and varied in document composition and page encoding. Capacity plans were included in only part of the material and were comparatively rare at page level. The quantitative dataset composition, processing split and computational environment used for the analysis are reported together in Section 4.1.

The document encoding also varied between projects. Some PDFs contained embedded text that could be extracted directly, while other files consisted mainly of rasterized pages or vector graphics without accessible text. In some cases, original CAD files were available and could be replotted as PDFs with embedded text. The replotting procedure used in AutoCAD 2026 is described in Appendix C.

1.2.2 Computational Environment

All source-document extraction, model execution, indexing and retrieval were performed locally on a laptop without dedicated GPU acceleration. The available environment affected computational feasibility. The computer used for the reported analysis and timing measurements is specified in Section 4.1.

1.2.3 Development Support

The project-specific software developed for the workflow was written in Python (Python Software Foundation, 2026). Codex and ChatGPT were used as development support during implementation (OpenAI, 2025, 2026). All generated code was reviewed, tested and adapted by the author before use in the experimental workflow.

1.3 Purpose

The purpose of this thesis is to determine whether AI-based techniques can support engineers in producing vessel-specific COW manuals from technical documents with varying formats, layouts and information content. The intended role is decision support: the system should locate, structure and present relevant evidence and assist with drafting, while the responsible engineer retains control of interpretation and approval.

In this thesis, the term AI-assisted refers to the complete workflow, which includes document processing, classification, computer vision, information retrieval and drafting. A large language model (LLM) is used specifically for questionnaire completion and manual drafting; these stages are therefore described as LLM-assisted.

1.4 Research Question

The purpose of the study is operationalized through one overarching research question:

Research question

To what extent can AI-based techniques support the production of COW manuals from CAD-derived drawings and related technical documents with varying formats, layouts and information content?

1.5 Limitations / Demarcations

The purpose of this thesis is not to produce a ready-to-deliver COW manual. Instead, the work is a pre-study intended to evaluate what this technology can currently accomplish and to identify its present limitations.

The study is therefore limited to assessing feasibility and to quantifying the accuracy of the extracted and structured information on the available material. The LLM-assisted drafting stage is a one-project proof-of-concept case study and is not an independent estimate of performance on new projects.

1.5.1 Data confidentiality and handling restrictions

The project was conducted under the confidentiality conditions defined in the thesis agreement with Scanjet (AB, 2025). These conditions restricted the disclosure and

external processing of confidential drawings, documents and project data. Confidential source documents and information capable of revealing complete drawings or system connectivity were therefore processed only within the Scanjet-controlled environment. This requirement shaped the system architecture: document interpretation was performed locally, while only minimized metadata and excerpts assessed as insufficient to reconstruct the source drawings or their connection logic could be supplied to the cloud-based LLM. All material included in this thesis was separately screened to exclude, anonymize or obtain approval for confidential information before submission.

2

Theory

This chapter presents theoretical and technical background on digital engineering documents and AI-based document analysis.

In addition to document-related background, the chapter introduces basic machine learning concepts to establish a common foundation for readers in marine technology. The methods used in the implemented workflow include direct PDF text extraction and OCR; TF-IDF, dictionary-based and fuzzy matching; LayoutLMv3 and multi-expert fusion; YOLO-based object detection; and a customized retrieval-augmented generation (RAG) workflow with an LLM agent for questionnaire completion and draft generation. The chapter also introduces file formats, engineering-document terminology, evaluation metrics, embeddings and multimodality as background needed to understand these methods. Related approaches that were not implemented, such as graph neural networks, are included where they help explain alternatives and possible future development.

2.1 Document Formats and Representations

2.1.1 File formats

Engineering-document workflows use different file formats for drawings, metadata and machine-readable instructions. The information may be represented as text, vector graphics, raster graphics or a combination of these forms.

Text formats Plain text is commonly encoded using Unicode UTF-8, which supports characters from different writing systems as well as special symbols (Wikipedia, 2026c). Text files can also contain syntax that provides additional structure. JSON Lines (JSONL) stores structured data as separate JSON records, normally with one record per line (Lines, 2026). Markdown uses lightweight markup to describe elements such as headings, lists and tables (Wikipedia, 2026b). YAML is a human-readable data-serialization format that represents mappings, lists and scalar values and is commonly used for configuration files and metadata (YAML Language Development Team, 2021). Although these formats serve different purposes, they are all text-based and can be read both by software and, to varying degrees, by humans.

Graphical formats Vector graphics represent graphical elements geometrically through lines, curves and shapes defined between points. Raster graphics instead represent an image as a matrix of colored pixels. Figure 2.1 illustrates how a raster

image is constructed from individual pixels and how each pixel stores separate red, green and blue values. Text and graphics are commonly treated as different modalities in document analysis.

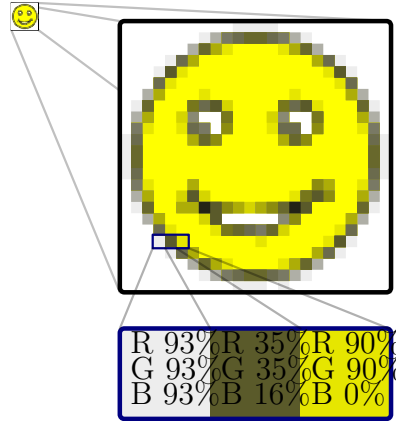


Figure 2.1: Raster image enlarged to show individual pixels and their RGB values.

Engineering-document formats DWG, derived from *drawing*, is a proprietary binary format used for storing two- and three-dimensional design data and metadata (Wikipedia, 2026a). It is associated with AutoCAD, and its graphical information is commonly stored in vector form (Corel, 2026). DXF (Drawing Interchange Format or Drawing Exchange Format) was developed by Autodesk to support the exchange of CAD data between different software and computing platforms (Wikipedia, 2026a; Wikipedia, 2026a). Like DWG, its graphical information is generally represented as vectors (Corel, 2026).

Portable Document Format (PDF), standardized as ISO 32000, is designed to present fixed-layout documents independently of application software, hardware and operating system. A PDF may contain embedded text, fonts, vector graphics and raster images (ISO, 2020; Wikipedia, 2026b). Visually similar PDF pages can therefore be encoded in different ways. This distinction is important in document analysis because embedded text can be extracted directly, whereas text stored as vector objects or raster images requires additional processing (ISO, 2020; Wikipedia, 2026b).

2.2 Engineering Documents

Piping & Instrumentation Diagrams (P&IDs) are engineering diagrams in which equipment, instrumentation and the flow between components are represented graphically, typically by standardized symbols and connecting lines (Bisht et al., 2023; ISO, 2014; Moreno-García et al., 2019). Figure 2.2 shows a simplified dual-pump arrangement and illustrates how pumps, valves, filters, instrumentation and connecting lines are combined to describe a process system.

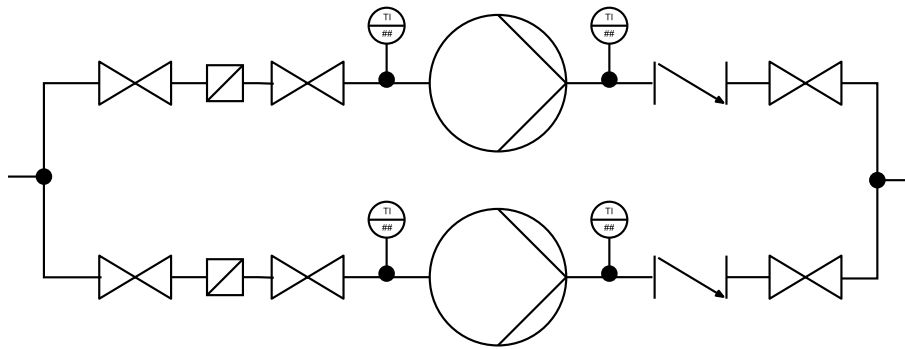


Figure 2.2: Simplified P&ID of a dual-pump arrangement, showing pumps, valves, filters and instrumentation.

These drawings are an industry-standard way of describing how a process system is constructed, but they are often information-dense and difficult to interpret. Equipment codes, line types, pipe dimensions, instrument tags and symbols are combined in a compact layout. Figure 2.3 provides an example of this information density in an engineering drawing, where repeated pumps, valves, equipment tags, pipe identifiers and connecting lines must be interpreted together. As a result, interpretation requires not only symbol recognition, but also text recognition, line detection and contextual understanding of how components are connected (Bisht et al., 2023; Elyan et al., 2020; Moreno-García et al., 2019; Pillai et al., 2024).

They are often produced in CAD software and distributed as printed drawings, CAD files or PDF files. Because of the amount of information they contain, paper sizes are commonly in the range A0–A3.

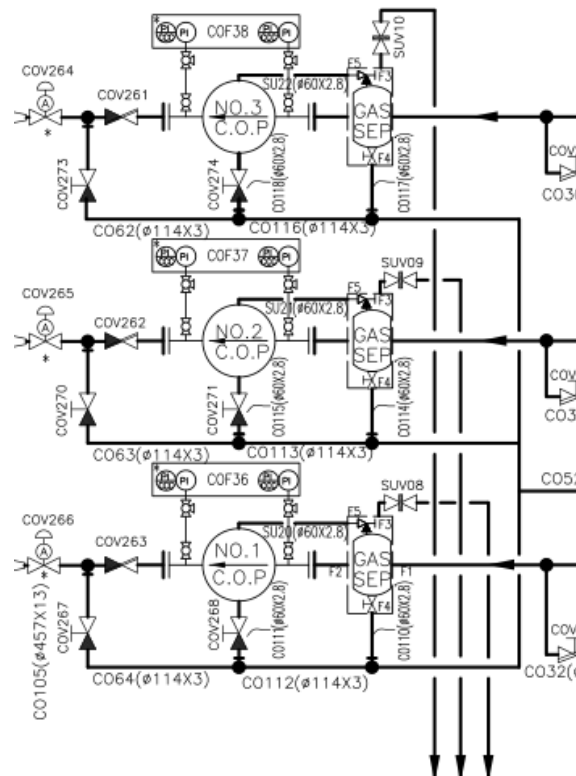


Figure 2.3: Pump configuration in an engineering drawing.

Equipment lists

Equipment lists accompany the P&ID drawings and provide structured information about the components represented in the drawing. Each component is normally connected to a reference number or tag in the P&ID, which corresponds to an entry in the equipment list. These entries may contain information such as size, material, pressure class, connection type, quantity and other relevant technical specifications.

Document codes

Document codes are used to identify and organize technical documentation in a consistent manner. In shipbuilding and marine engineering projects, such codes may indicate the system, discipline, document type, revision status or project-specific document number. This makes it possible to trace drawings, equipment lists, manuals and other technical documents across different project phases and between different stakeholders.

A work breakdown structure divides project scope into hierarchical components and supports the organization and control of project information (International Organization for Standardization, 2018). In shipbuilding documentation, International Work Breakdown Structure (IWBS) can be used as a collective term for hierarchical coding structures that divide a vessel or project into progressively more specific systems, work packages or document groups. An IWBS-related code can therefore help indicate where a drawing or document belongs within the overall project structure.

The exact code levels and their meanings depend on the coding system used by the relevant organization or project.

The SFI Group System is one example of such a coding system. SFI is a function-oriented classification system that groups vessel systems, equipment, machinery and related information according to function. It can support structured handling of specifications, estimates, purchasing, maintenance, drawings and other project information (Urke, 1976).

2.2.1 Standards

Technical drawings and equipment lists commonly refer to standardized designations for pipework components, such as DN and PN. These designations provide compact information about component size and pressure classification, and are important for interpreting valves, flanges, pipes and other fittings in P&ID drawings and equipment lists.

DN

DN, or nominal size, is a standardized designation used for pipework components. It consists of the letters DN followed by a dimensionless number and is used as a reference size for components such as pipes, valves, flanges and fittings (International Organization for Standardization, 1995). DN should not be interpreted as an exact internal or external diameter, since the actual dimensions depend on the relevant pipe or component standard, material and wall thickness. Instead, DN provides a common reference designation that helps ensure compatibility between connected pipework components.

PN

PN, or nominal pressure, is a standardized pressure designation used for pipework components. It consists of the letters PN followed by a dimensionless number and is used to classify components according to pressure-related mechanical and dimensional properties (European Committee for Standardization, 2006). The PN value is therefore a reference designation rather than a complete statement of the allowable working pressure under all operating conditions. The actual allowable pressure depends on factors such as material, temperature, component type and the applicable product standard.

2.3 AI/ML Methods

Artificial intelligence (AI) is used as a broad term for computational methods that perform tasks involving interpretation, prediction or decision-making. Machine learning (ML) is a branch of AI in which a system improves its performance for a given task based on training experience (Mitchell, 1997). In technical-document analysis, machine-learning models can be used to recognize text, detect visual objects, classify content and combine information from multiple sources.

Machine-learning models learn from collections of examples. In natural language processing, a collection of documents or texts used for analysis, training or evaluation is commonly called a corpus (Jurafsky & Martin, 2026). Before text can be processed by a model, it is divided into smaller units called tokens. Depending on the tokenization method, a token may represent a word, part of a word, a character or punctuation (Jurafsky & Martin, 2026). In semantic labelling, each token is assigned to a defined class. Tokens that do not belong to any target class are assigned to the outside, or O, class. During model training, one complete pass through the training dataset is called an epoch (Google Developers, 2026).

The information supplied to a machine-learning model must be represented numerically. An embedding maps complex information, such as words or their contexts, to numerical vectors in a lower-dimensional space (Bengio et al., 2003; Zhao et al., 2026). Images, signals and sequences are examples of Euclidean data because their elements are arranged in regular structures with consistent local relationships. This differs from graph-structured data, where relationships may connect elements without following a regular geometric arrangement (Bronstein et al., 2017). Document-processing systems also use metadata to describe and organize files, pages, detected objects and extracted information (Riley, 2017). Such metadata can be stored in machine-readable text formats such as JSONL (Lines, 2026).

2.3.1 Evaluation Metrics

Performance in classification and labelling tasks can be described using a confusion matrix and metrics such as accuracy, precision, recall and F1-score (Powers & Ailab, 2020; Sokolova & Lapalme, 2009). Table 2.1 shows the four possible outcomes when one class is evaluated against all other observations.

Actual class	Predicted class	
	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Table 2.1: Binary confusion matrix for one class evaluated against all other observations.

For one evaluated class, the entries of a confusion matrix are defined as follows:

- true positive (TP): an item belonging to the class is correctly assigned to it;
- false positive (FP): an item not belonging to the class is incorrectly assigned to it;
- true negative (TN): an item not belonging to the class is correctly rejected; and
- false negative (FN): an item belonging to the class is incorrectly rejected.

Accuracy is the proportion of all predictions that are correct:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.1)$$

Precision describes how many of the items assigned to a class actually belong to that class:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (2.2)$$

Recall describes how many of the items belonging to a class are identified:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (2.3)$$

The F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \quad (2.4)$$

Accuracy provides a clear overall result, but it can give an incomplete picture when some classes contain many more examples than others. A confusion matrix complements accuracy by showing which classes are confused with each other.

For a problem with C classes, macro-F1 is the unweighted mean of the class-wise F1-scores:

$$\text{Macro-F1} = \frac{1}{C} \sum_{c=1}^C F1_c. \quad (2.5)$$

Balanced accuracy is the unweighted mean of recall over the C classes:

$$\text{Balanced Accuracy} = \frac{1}{C} \sum_{c=1}^C \text{Recall}_c. \quad (2.6)$$

Macro-F1 and balanced accuracy therefore give each class equal influence, regardless of how many examples it contains. Reporting the number of examples in each class also helps indicate how stable the class-level results are (Powers & Ailab, 2020; Sokolova & Lapalme, 2009).

For object detection, precision and recall describe how well a model locates reference objects or regions. In information retrieval, the same terms describe the relevance and coverage of the returned results. The underlying unit of evaluation therefore needs to be stated when these metrics are interpreted.

2.3.2 Text-Based Machine Learning Approaches

Text-based machine learning approaches operate on textual data that has been transformed into numerical representations. Text has a naturally sequential structure, where information is represented as ordered units such as characters, words, subwords or tokens, making it suitable for established methods in feature extraction, classification and sequence modelling. Figure 2.4 illustrates this sequential structure, with each node representing one text unit and the arrows indicating the order between the units. Earlier approaches commonly used word counts, term frequencies or weighted term representations to learn statistical relationships between words and document categories, and text classification became one of the early areas where machine learning showed practical usefulness for tasks such as document categorization, information filtering and automatic organization of large document collections (Sebastiani, 2002).

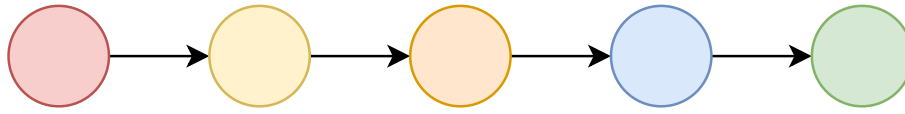


Figure 2.4: Sequential representation of text units.

Semantics Semantics deals with the meaning of linguistic expressions, including words and sentences. In natural language processing, this meaning-related information is often operationalized through word labeling, where each token in a sequence is assigned a label representing a syntactic, semantic, or task-specific category. This sequence labeling paradigm is widely used in tasks such as named entity recognition and information extraction (Encyclopedia, 2026).

Large Language Models

Large language models (LLMs) are neural language models trained on large text corpora, typically with self-supervised objectives, and most modern LLMs are based on the Transformer architecture (Vaswani et al., 2023; Zhao et al., 2026). Natural-language instructions and examples allow a single model to support tasks such as question answering, summarization, drafting, translation and code assistance. Large-scale language models have demonstrated broad zero-shot and few-shot task adaptation (Brown et al., 2020), and structured prompting can improve performance on tasks that involve intermediate reasoning steps (Wei et al., 2022).

An important property of LLMs is their flexibility in combining several pieces of information, accounting for context and proposing relationships, inconsistencies or implications that may not have been specified in advance. For example, a model may suggest that several tank groups represent a segregation arrangement or identify a conflict between equipment values stated in different documents. Such suggestions are not established engineering facts. A useful distinction can therefore be made between an extracted fact, which is directly supported by a source location, a candidate inference, which states its premises and requires engineering confirmation, and an unresolved issue, for which the available evidence is insufficient.

This flexibility does not make an LLM authoritative. A generated explanation can appear plausible without faithfully representing why a conclusion was produced (Turpin et al., 2023). Retrieval-augmented generation addresses part of this limitation by conditioning generation on inspectable external evidence rather than relying only on model parameters (Lewis et al., 2020). In engineering applications, factual statements and model-proposed inferences should therefore remain traceable to their supporting sources and subject to expert review.

Agents

An AI agent is a software system that can autonomously perform tasks on behalf of a user or another system by perceiving information, making decisions, and taking actions to achieve a goal (IBM, 2026).

Code generation

Code generation is one practical application of LLMs in which the model is used to produce, complete, modify or explain source code based on natural-language instructions or existing code context.(Chen et al., 2021).

TF-IDF

Term frequency–inverse document frequency (TF-IDF) is a standard weighting approach in information retrieval (Jones, 2004; Manning et al., 2009; Salton & Buckley, 1987). It is a relatively simple but still highly effective technique to distinguish one specific type of document inside a large corpus of documents.

Based on the following calculations:

Term frequency: counts how many times term t occurs in document d .

$$\text{tf}(t, d) = f_{t,d} \quad (2.7)$$

Document frequency: number of documents in the corpus D that contain term t at least once.

$$\text{df}(t) = |\{ d \in D : t \in d \}| \quad (2.8)$$

Inverse document frequency: down-weights terms that appear in many documents and up-weights rarer terms.

$$\text{idf}(t, D) = \log \left(\frac{|D|}{\text{df}(t)} \right) \quad (2.9)$$

TF-IDF: combines term frequency and inverse document frequency to score term importance for document d in corpus D .

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \text{idf}(t, D) \quad (2.10)$$

Shakespeare example Inverse document frequency (IDF) captures the intuition that a term is more informative if it occurs in only a few documents, and less informative if it appears in many or all documents of the corpus. The following hypothetical corpus of 37 plays illustrates the calculation using example document frequencies. Words that occur in only one or a few plays, such as *Romeo* or *Falstaff*, are highly discriminative, while words that appear in nearly every play, such as *good* or *sweet*, provide little help for distinguishing between plays.

Using the common definition

$$\text{idf}(t, D) = \log_{10} \left(\frac{|D|}{\text{df}(t)} \right), \quad (2.11)$$

Table 2.2 presents the corresponding document frequencies (df) and IDF values for the example corpus.

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.966
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

Table 2.2: Example df and idf values for words across Shakespeare’s 37 plays.

Rapidfuzz

RapidFuzz is a high-performance library designed for fuzzy string matching and text similarity comparison. It can be used to measure how closely two pieces of text resemble each other, even when the words are not identical. Fuzzy matching is useful for handling OCR token errors, where a single character may be misread, for example when 0 and 0 have similar visual appearance but different interpretation.

2.3.3 Computer Vision

Computer vision is a field of artificial intelligence concerned with enabling computers to extract, interpret and use meaningful information from images and visual documents (Huang et al., 2022; Moreno-García et al., 2019; Redmon et al., 2016). In the context of engineering documents, this includes symbol detection, OCR, layout interpretation and connectivity analysis. Figure 2.5 illustrates the regular two-dimensional neighbourhood structure of a raster image, in contrast to the one-dimensional text sequence shown in Figure 2.4.

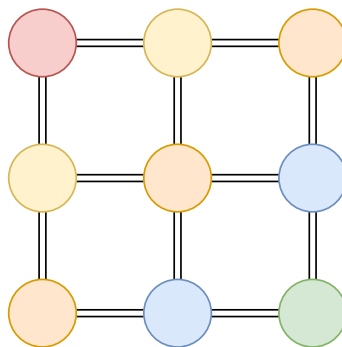


Figure 2.5: Regular grid representation of a raster image.

YOLO - You only look once

YOLO is an object detection method that predicts object locations and class probabilities directly from an image in a single evaluation (Redmon et al., 2016). Its main advantage is speed, since the full image is processed in one pass. The input data are raster images.

OCR

Optical character recognition (OCR) is the task of converting images of printed or written text into machine-readable text (Smith, 2007). When machine-readable text cannot be recovered directly, OCR can be applied to raster images or to vector-based graphics after rasterization. The conversion is not exact and can be disturbed by lines or nearby symbols in dense drawings. Text in engineering drawings can also occur at several 90-degree rotations. More importantly, OCR recovers visual character content rather than the original CAD representation: it does not preserve layers, vector-object semantics, component identities or piping connectivity.

2.3.4 Multimodal Model Architectures

Multimodality in machine learning refers to the use of data from multiple modalities, such as text, images, audio, video, sensor signals or structured metadata. Each modality represents information in a different form, and multimodal models aim to combine or relate these sources in order to improve understanding, prediction or decision-making. This is particularly useful when one modality alone does not contain all relevant information. In technical-document analysis, for example, embedded text may describe component names or dimensions, while drawings, tables and spatial layout provide additional visual and structural context. Multimodal machine learning is therefore relevant when information must be interpreted across several complementary data types (Baltrušaitis et al., 2019).

LayoutLMv3

LayoutLMv3 is a multimodal Transformer model for Document AI that jointly learns from text and image information and is intended for both text-centric and image-centric document understanding tasks (Huang et al., 2022).

Multi-Expert Fusion

In classification problems, a single model is often not sufficient to capture all relevant aspects of the data. Different models may be strong for different reasons: one may learn visual or spatial structure, another may represent domain rules, and a third may capture repeated terminology. The general idea of *ensemble learning* is therefore to combine several predictors in order to obtain a final decision that is more robust than any single predictor alone (Dietterich, 2000).

The expert outputs can be combined in a fusion layer, and the relative contribution of each expert can be learned from data. A common ensemble strategy is *late fusion*, also called *decision-level fusion*. In late fusion, each predictor first produces

its output independently, after which the outputs are combined in a separate fusion step. This allows the predictors to specialize in different types of information while contributing to a common final decision (Kittler et al., 1998).

Assume that an input item can belong to one of C possible classes and that the ensemble contains E experts. Each expert e produces a score vector of length C ,

$$\mathbf{s}^{(e)} \in \mathbb{R}^C, \quad e = 1, \dots, E, \quad (2.12)$$

where each element represents one expert's support for a class. The scores may originate from different model types or information sources.

The simplest late-fusion model forms a weighted sum of the expert outputs:

$$\mathbf{z} = \sum_{e=1}^E \lambda_e \mathbf{s}^{(e)}, \quad (2.13)$$

where each λ_e is a scalar weight describing the overall contribution of expert e . The final predicted class is the class with the highest fused score:

$$\hat{y} = \arg \max_c z_c. \quad (2.14)$$

Here, $\arg \max_c z_c$ denotes the class index c corresponding to the largest fused score. A more flexible approach is to let the weights depend on the class. Instead of one scalar weight per expert, the model learns a full matrix of class-specific weights (Cao et al., 2015):

$$W \in \mathbb{R}^{C \times E}. \quad (2.15)$$

The fused score for class c can then be written as

$$z_c = \sum_{e=1}^E W_{c,e} s_c^{(e)} + b_c, \quad (2.16)$$

where b_c is a class-specific bias term. This formulation means that each class can rely on the experts differently.

A further extension is to make the expert weights depend not only on the class but also on the current input. A gating function can produce input-specific correction factors. In simplified form, this can be written as

$$\mathbf{g}(\mathbf{x}) = 1 + \alpha \tanh(f_\theta(\mathbf{x})), \quad (2.17)$$

where $\mathbf{x}$ is a feature vector representing the input and its context, f_θ is a learned function and α limits how strongly the gate may change the base weights. The final fused score then becomes

$$z_c = \sum_{e=1}^E g_e(\mathbf{x}) W_{c,e} s_c^{(e)} + b_c. \quad (2.18)$$

The class-specific matrix W represents the model's general weighting of each expert for each class, while the gate $\mathbf{g}(\mathbf{x})$ adjusts those weights for an individual input. This combines global weighting with local adaptability. The gating idea is related to the broader theory of expert combination, in which a second model learns how strongly different experts should contribute under different conditions (Jacobs et al., 1991).

Graph Neural Networks

Graph neural networks (GNNs) are neural models designed for graph-structured data, where entities are represented as nodes and their relationships as edges. Graphs are a highly general representation because they can capture complex relational structure that is difficult to express in regular data formats such as images or sequences. This makes GNNs particularly relevant for P&ID drawings, since such diagrams describe interconnected process equipment, piping, and instrumentation, which can naturally be interpreted as a graph. At the same time, graph data is less regular than Euclidean data, making graph-based learning more challenging and often more computationally demanding than conventional approaches. Figure 2.6 illustrates an irregular graph in which nodes can have different numbers and directions of connections, unlike the regular sequence and image-grid structures shown previously.

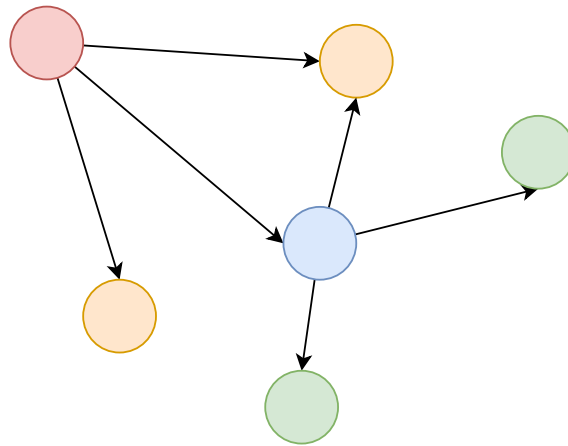


Figure 2.6: Example of an irregular graph with nodes and directed edges.

3

Methods

This chapter describes the method used to generate a draft of Section 3.2, *Cargo Oil System*, for a Crude Oil Washing (COW) manual. The method uses a customized retrieval-augmented generation (RAG) workflow to capture domain-specific knowledge and enable fast retrieval of relevant information on a local machine. Local processing was required because the source documents contain confidential engineering information. Complete drawings, full-page images, full OCR output, spatial layouts, CAD-layer information and system-connectivity information were therefore retained within the Scanjet-controlled environment.

The workflow first categorizes the shipyard documentation at page, token and region level and stores the outputs locally as structured metadata. Retrieval then selects the facts and bounded evidence needed for a manual drafting task. Only selected metadata and local excerpts assessed as insufficient to reconstruct a drawing or its connection logic were eligible for transfer to the cloud LLM. The LLM organized this minimized evidence into a draft for engineering review. The resulting deployment is a controlled hybrid architecture, not an entirely local pipeline.

The outputs from the local processing stages were progressively combined into an enriched page-wise JSONL representation referred to in this thesis as Final Embeddings. The following sections describe the individual components added to this representation. Its complete structure and its use as the common input to information-region retrieval are presented in Section 3.7.

As stated in Section 1.4, the thesis investigates to what extent AI techniques can support COW manual production. The workflow uses machine-learning models trained on labelled data together with LLM-assisted synthesis.

Technical alternatives were evaluated separately for each stage of the workflow. Confidentiality and data minimization were treated as mandatory constraints: methods requiring unrestricted transfer of drawings or system-level information were not considered eligible. Among the eligible alternatives, the selection considered task-specific performance, computational cost, practical feasibility and traceability. Each selection is explained in the section where the method is introduced.

The overall sequence from source documents through local processing, controlled evidence transfer, LLM-assisted drafting and engineering review is summarized in Figure 3.1. The following sections describe the individual processing stages.

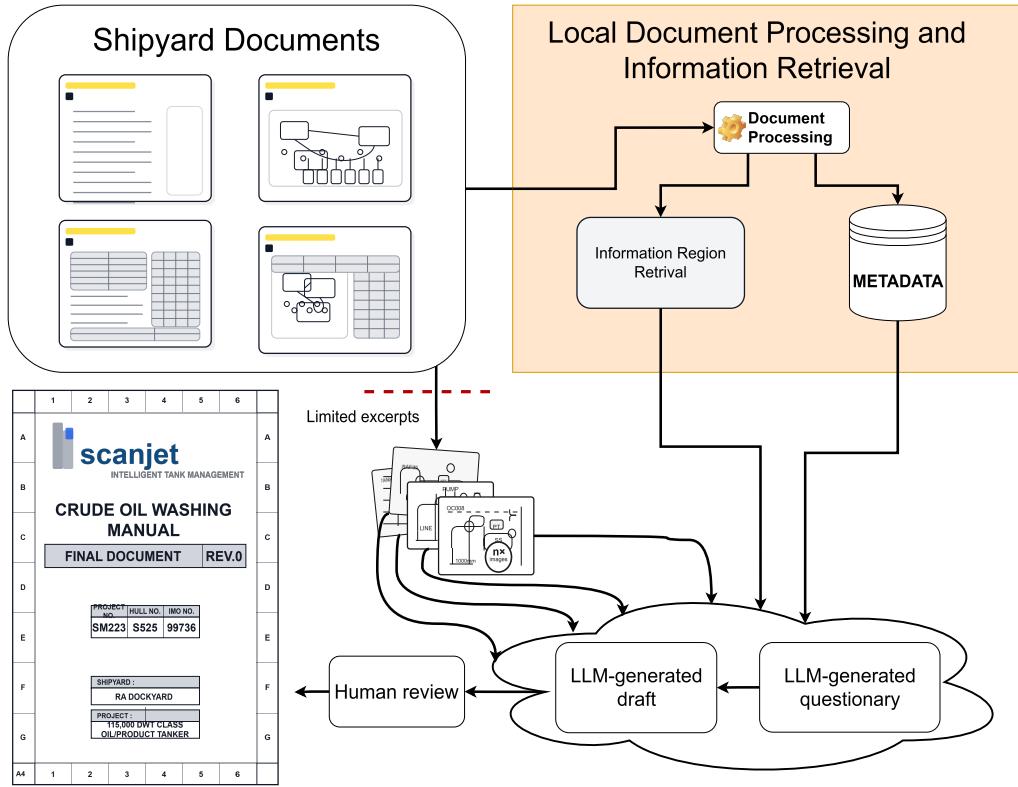


Figure 3.1: Overview of the workflow from shipyard documents through local document processing and controlled evidence transfer to LLM-assisted questionnaire completion, manual drafting and engineering review.

3.1 Local Document Processing and Information Retrieval

The local document processing and information retrieval block in the overall workflow is expanded in Figure 3.2. It shows how the shipyard documents pass through text extraction, page categorization, semantic labelling, page-area classification and information-region detection. The resulting page- and text-level information is combined into structured metadata and used for information-region retrieval before selected evidence is supplied to the LLM-assisted stage. The following sections describe these local processing steps individually.

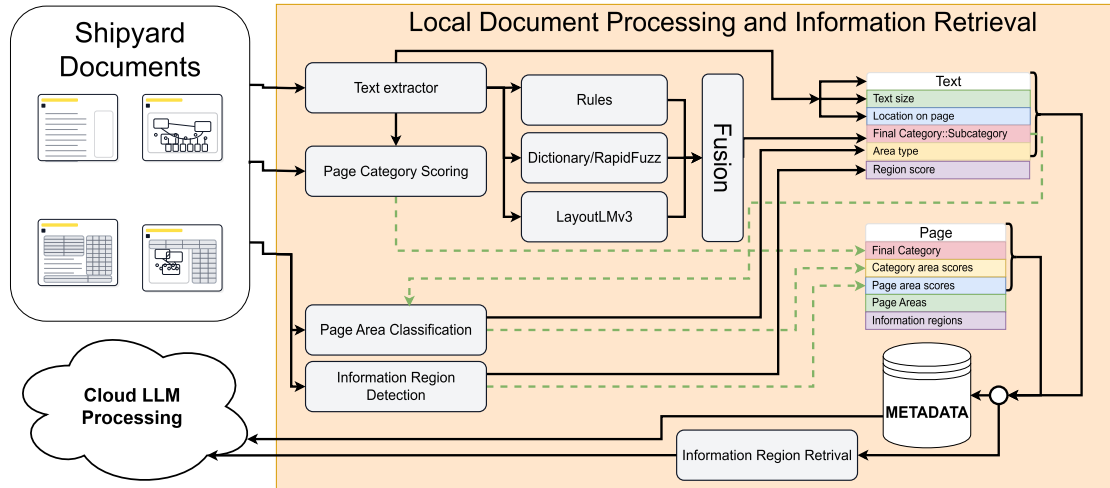


Figure 3.2: Expanded view of the local document processing and information retrieval block in the overall workflow.

3.2 Text Extraction

Text extraction followed a hierarchy intended to preserve source information while limiting recognition errors and computational cost. Usable embedded text was extracted directly from the PDF. When native CAD files were available, they could be replotted to produce PDFs with machine-readable text. OCR was applied when usable embedded text was absent and could also be forced when the direct result was incomplete or structurally unreliable. On mixed pages, regions containing embedded text were masked during rendering before OCR, while the original embedded text was retained. This prevented the same text from being extracted twice.

The OCR engine was selected through a practical comparison of Tesseract, licensed local ABBYY software and the Azure OCR container. The locally executed Azure implementation produced the strongest result in this comparison and was therefore used in the workflow. The comparison was a tool-selection experiment rather than a standardized OCR benchmark. PaddleOCR and EasyOCR were not configured or evaluated, so no performance comparison is made with these alternatives. The selected Azure container ran locally through Docker; “Azure” identifies the OCR technology and container image rather than use of a cloud OCR API. Complete page images and recognized text therefore remained within the controlled local environment.

Direct extraction with a multimodal LLM was also considered. This approach could interpret the complete page in one step and thereby avoid error propagation between separate OCR, page-categorization, token-labelling, region-detection and retrieval stages. A cloud implementation would, however, require complete page content to leave the Scanjet-controlled environment. A local implementation would preserve confidentiality, but suitable hardware for running a sufficiently capable multimodal LLM was not available during the project. Direct LLM extraction was therefore not feasible.

The extraction stage also established the stable page coordinate system used by

the later models. It produced the base page-wise JSONL representation that was progressively enriched by the subsequent processing stages rather than requiring those stages to operate directly on the original PDF files. Each extracted word has a normalized position, size and page association.

Output The output was one JSONL file per page, containing one `page_meta` record followed by one record per detected word or text segment. The metadata record stored page dimensions, page rotation and coordinate-space information, while each text record stored the extracted text, bounding boxes and additional geometric information. These files provided the base records used by page-category scoring, semantic labelling and the later metadata pipeline; the common coordinate system also allowed text and region predictions to be related to the original page.

3.3 Page Category Scoring

Given the required drawing types and sections stipulated in MARPOL Annex I and the COW manual requirements (International Maritime Organization, 1979, 2004), the first broad categorization of the document corpus was to classify each page into one of the document categories used for retrieval. Correct page categorization reduces the search space for later extraction, since many facts are strongly associated with a specific document type. For example, information about cargo pumps should normally be found within the cargo oil system material. The page categories are presented in Table 3.1, together with the instances in the corpus from Batches 1–4 described in Appendix D. A basic labelling tool was developed to label all pages in the corpus.

Table 3.1: Page-category classes.

Category	Training instances
Cargo Oil System	311
Inert Gas and Tank Ventilation Systems	184
Water Ballast System	181
Tank Washing System	112
Upper Deck Plan	81
Capacity Plan	15

A multi-source TF-IDF method was selected because it was practical for the available domain-specific corpus, could run locally on the CPU-only computer and kept the contribution from each text source inspectable. An end-to-end neural page classifier was not implemented because it would require additional labelled data and computation; no performance comparison with such a classifier is therefore made. Separate TF-IDF scores were calculated from the full-page text, extracted title text and file name (Jones, 2004; Manning et al., 2009; Salton & Buckley, 1987). The title text consisted of tokens assigned the `Title::Title` label by the semantic labelling model. When the scores were inconclusive or supported multiple categories,

a page-sequence fallback used neighbouring pages to resolve the classification. The TF-IDF representation was fitted statistically rather than through iterative neural optimization. The configured method was evaluated on the 67-page validation set and exported the selected page category together with the intermediate score vectors.

Output The output was a final page category together with the intermediate score vectors from the text sources and the combined decision. These values were added to the page-level JSONL metadata. Final Embeddings retained both the selected category and the score vectors, while information-region retrieval used the category to restrict searches and apply category-specific rules.

3.4 Word Semantic Labelling

The semantic label taxonomy was used as an explicit way of transferring the author’s domain knowledge into the system. The choice of classes and the text elements assigned to them reflected which equipment, properties, units and document features the author judged relevant to later retrieval and COW manual drafting. This made the intended categories inspectable, but it also meant that concepts not anticipated when the taxonomy was designed were not represented explicitly. A fusion of rules, fuzzy dictionary matching and LayoutLMv3 was selected because the three methods use different evidence. Rules capture recurring engineering patterns and can respond to unseen values, the dictionary recognizes known terminology despite minor OCR variation, and LayoutLMv3 adds page position and surrounding layout. The late-fusion model was used instead of relying on a single branch so that these signals could be combined while the contribution from each branch remained available for inspection. Figure 3.3 summarizes how the three branches are combined.

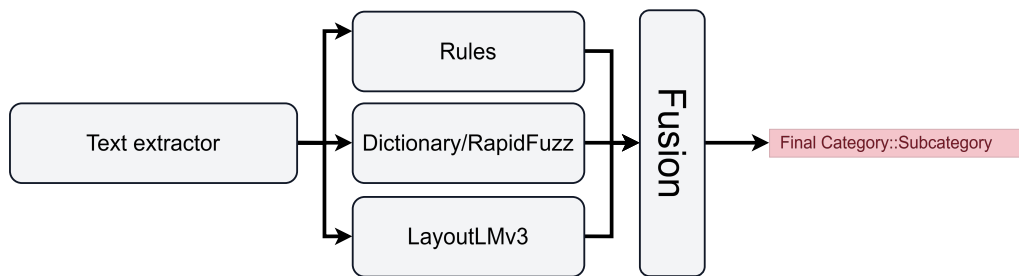


Figure 3.3: Semantic-labelling workflow combining rules, dictionary-based fuzzy matching and LayoutLMv3 through late fusion.

A more general end-to-end model, such as a capable multimodal LLM, was considered as an alternative for assigning semantic meaning directly from page content. Such a model might identify useful patterns beyond the predefined taxonomy and reduce reliance on author-defined classes. However, processing complete pages with a cloud model was outside the approved confidentiality boundary, and the hardware required to run such a model locally was not available during the project. This approach would also have made it more difficult within the project to determine

which technical conventions the model recognized reliably. For example, recognition of the different notations used for pipe dimensions could not be assumed without a dedicated evaluation.

The word labelling stage assigns semantic meaning to individual words and phrases by labelling them with categories and subcategories, such as equipment, compartments, units, title elements and pipe descriptions. The largest individual category was the background class O, since many tokens did not belong to any of the defined semantic categories. This data can later be used to extract or find specific groups of information from the corpus. The labelling process was developed iteratively. Initial pages were labelled manually in broad classes. These labels were then refined into more specific subcategories, and custom tools were developed to speed up later labelling rounds by suggesting labels. As the model improved, later batches could be labelled more efficiently by correcting predictions instead of starting from scratch for every page. Batches 1–3 in Appendix D were used for training, validation and test data, with the resulting token-label distribution reported in Table 3.2.

Table 3.2: TextModel token labels.

Token label	Training instances	Validation instances	Test instances
O	53666	6073	3496
Equipment::Valve	12050	1906	1154
Pipe::Dimension	8500	1974	1564
Pipe::Misc	8830	1907	595
Equipment::Equipment	4159	1149	611
Units::Geometric	3412	518	186
Compartments::Cargo tanks	2866	410	373
Compartments::Ballast tanks	1960	366	125
Compartments::Compartments	1539	416	331
Units::Number	1802	157	62
Units::Pressure	1737	243	13
Units::Mechanics	957	548	32
Units::Units	1083	83	26
Title::Title	1002	95	87
Equipment::Cargo pump	739	241	75
Equipment_type::Equipment type	779	170	34
Compartments::Slop tanks	660	120	99
Compartments::Misc tanks	436	310	21
Equipment::Tank cleaning	496	150	82
Units::Capacity	436	166	91
Medium::Medium	507	122	48
Compartments::Fuel tanks	275	194	31
Equipment::Ballast pump	316	64	29
Equipment::Cargo heating	181	91	32
Equipment::Cargo stripping	257	4	41
Equipment::Misc pump	195	60	37
Compartments::Cofferdam	166	58	16

Token label	Training instances	Validation instances	Test instances
Units::Misc	154	54	1
Segregation::Segregation	105	7	41
Title::SFI code	78	6	9
Equipment::Slop pump	61	28	0
Equipment::Ballast stripping	54	11	19

3.4.1 Rules

The first expert is a rule-based categorization engine. It classifies extracted text strings by matching them against predefined patterns stored in an external rules file. The rules are designed to capture recurring technical expressions and naming conventions found in the project documents, such as DN and PN standard values, tank combinations, valve labels, equipment labels and other domain-specific identifiers.

The rule file is intentionally made extendable. New categories, subcategories and matching patterns can be added without changing the source code of the categorization engine. To make the rules more compact and reusable, the format supports placeholders and wildcard symbols. Placeholders are used to define reusable groups of values, such as valid valve prefixes or nominal pipe sizes, while wildcard symbols allow patterns to match variable parts of a string. This makes it possible to express many related text patterns using relatively few rules.

Excerpt from rules file

```
# Format per rule (TAB separated):
#   category<TAB>subcategory<TAB>pattern
#
# Pattern symbols:
#   L = any letter (A-Z / a-z)
#   N = any digit (0-9)
#   # = numeric token (digits with
#       optional '.' or ',' separators)
#   $ = any text (zero or more chars)
#   %Name% = placeholder token (define with %Name%=A|B|C)

%ValvePrefix%=AC|AD|AE|AR|AQ|B|BA|BAF|BAV|BB|BF|BW|BWF
%DN%=6|8|10|15|20|25|32|40|50|65|80|90|100|115|125|150

Pipe      Dimension      %ValvePrefix%#-$-%DN%A
```

Output For each extracted token, the rules branch produced a category-probability vector based on the matched patterns. This vector was passed to the fusion layer as one of the three expert inputs.

3.4.2 Dictionary

The dictionary expert is a trained database with known strings and words matched to the words in the document. It uses RapidFuzz fuzzy matching to overcome small differences in how the text is formatted and to mitigate OCR detection errors. The dictionary uses a statistical model similar to TF-IDF that allows it to give a strong signal when a unique word is connected to a category.

Output For each extracted token, the dictionary branch produced a category-probability vector based on the known terms and their fuzzy-match scores. This vector was passed to the fusion layer as the second expert input.

3.4.3 LayoutLMv3

As the third expert, a LayoutLMv3 model was trained using the labelled data. Since LayoutLMv3 is a multimodal model, it can learn patterns related to where on the page and in what neighbourhood a specific type of text usually exists. The clearest example of this is the `Title::Title` category, which represents the title text of a page. This text usually appears at the sides of the page in a table-like title block, all of which are clues the model can use. The base training configuration is summarized in Table 3.3.

Table 3.3: Base LayoutLMv3 training configuration.

Parameter	Value
Pretrained checkpoint	microsoft/layoutlmv3-base
Task	Token classification
Architecture	LayoutLMv3ForTokenClassification
Number of labels	32 (including background label 0)
Max sequence length	512
Image size	224×224
OCR inside LayoutLMv3	Disabled
OCR input source	External OCR words and bounding boxes supplied to the processor
Only first subword labeled	Enabled
Epochs	8
Learning rate	5×10^{-5}
Weight decay	0.01

Output For each token processed by the model, the LayoutLMv3 branch produced a category-probability vector using the token text, page position and surrounding layout. This vector was passed to the fusion layer as the third expert input.

3.4.4 Fusion

The three experts all have their individual strengths. Rules can react to data not seen before if it follows a specific pattern. They also give a strong signal for exact or near-exact matches. The dictionary branch can learn quickly from observed examples, and RapidFuzz allows close matches and OCR errors to be mitigated. The multimodal design of LayoutLMv3 allows it to learn labels in relation to page location and surrounding layout structure. These three experts are merged in an ML fusion layer that learns when to trust each expert and adds a token-specific component.

Output The fusion stage produced one final category and subcategory for each token, a combined probability vector and the individual probability vectors from the rules, dictionary and LayoutLMv3 branches. These values were added to the corresponding text records in the page JSONL files. The title labels were used by page-category scoring, and the complete semantic labels and score vectors were retained in Final Embeddings for information-region retrieval.

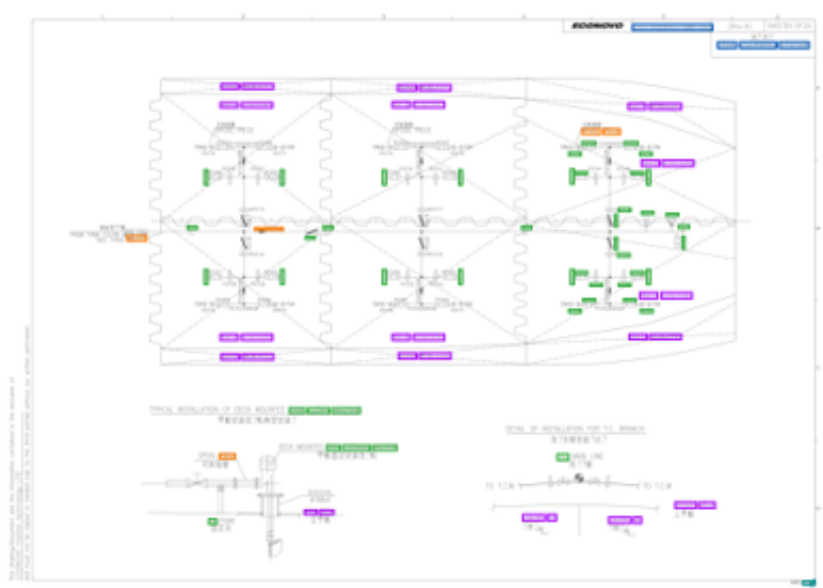
3.5 Page Area Classification

YOLOv8n was selected for page-area classification because text, tables, drawings and piping diagrams are distinguished partly by their spatial extent and visual layout. An object detector provides both a class and a bounding box, which was required for later filtering, while the nano architecture limited the computational cost. A text-only method was not considered sufficient because similar words can occur in several page structures. Full-page inference remained local.

A YOLO model was trained on full-page images to detect the general data structure in the document, categorizing areas as text, table, drawing or piping diagram. This is important because certain types of data are usually present only in specific page structures. It also becomes important in projects where many equipment table pages are included, since those pages can otherwise create costly and unnecessary processing when locating specific data. To increase accuracy, especially in the separation between the drawing and piping diagram classes, the training set contained augmented coloured boxes representing the main text-label category of each text box. These visual cues are useful because, for example, piping diagrams often include tanks in rows, while drawings contain dimension metrics. The deployed training settings are summarized in Table 3.4, and Figure 3.4 shows an example of the augmented training representation.

Table 3.4: Training settings for the page-content YOLO detector.

Setting	Value
Base architecture	Ultralytics YOLOv8n.
Initial pretrained package	yolov8n.pt.
Input image size	1024 × 1024 pixels.
Epoch budget	100.
Batch size	4.
Patience	100 epochs.

**Figure 3.4:** Example training page with semantic text-label categories overlaid as coloured boxes.

Batches 1–3 in Appendix D were used to train the model. The resulting class distribution is reported in Table 3.5.

Table 3.5: Page-content YOLO classes.

Class	Training instances	Validation instances
Table	574	144
Drawing	438	119
Text	218	46
Piping diagram	213	45

Output The page-area classifier produced bounding boxes with an area class and confidence score for text, table, drawing and piping-diagram regions. The detections and page-level area summaries were added to the page JSONL files. They were then included in Final Embeddings and used by the information-region detector and

retrieval logic to filter or interpret candidates according to the surrounding page structure.

3.6 Valuable information regions

Regions containing information considered relevant for producing the manual were labelled according to the categories shown in Table 3.6. A complete pass of all pages was performed, and any region judged by the author to contain potentially useful manual evidence was marked. These regions form the main evidence base for the final manual draft. Figure 3.5 shows representative labelled regions containing a ballast-pump arrangement, a cargo-segregation diagram and a ballast-equipment table.

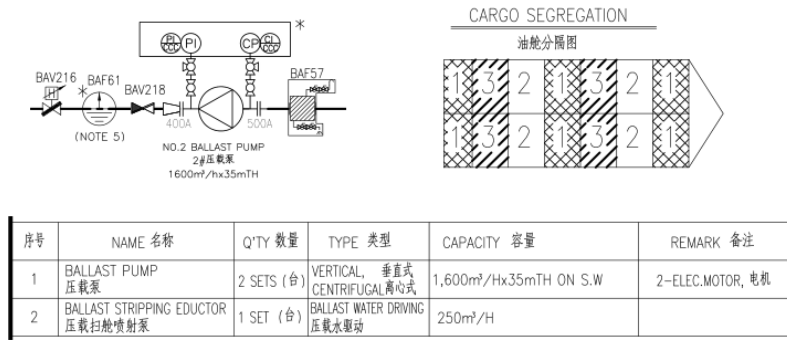


Figure 3.5: Examples of labelled information regions: ballast-pump arrangement, cargo segregation and ballast-equipment data.

Table 3.6: AreaScore classes.

Class	Training instances	Validation instances
Cargo::Pump	240	77
Cargo::Heating	98	49
Ballast::Capacity	80	3
Inert Gas::PV Valve	75	21
Ballast::Pump	72	17
Cargo::Capacity	64	4
Cargo::Stripping	46	10
Ballast::Stripping	35	7
Tank Cleaning::Heater	33	6
Cargo::Segregations	20	3
Tank Cleaning::Pump	18	4
Tank Cleaning::Machine	16	4
Inert Gas::Capacity	6	1

3.6.1 Information Region Detection

YOLOv8n was also selected as the initial detector for valuable information regions because many target regions have recurring visual arrangements and require a spatial output. The detector was used to propose candidate regions rather than as the only retrieval method, since visual appearance alone could not reliably distinguish information contained in ordinary text and tables. The labelled training and validation instances for its target classes are reported in Table 3.7.

Table 3.7: LabelFinal YOLO classes.

Class	Training instances	Validation instances
Cargo::Pump	298	19
Cargo::Heating	146	1
Inert Gas::PV Valve	86	10
Ballast::Capacity	81	2
Ballast::Pump	76	13
Cargo::Capacity	66	2
Cargo::Stripping	45	11
Ballast::Stripping	33	9
Tank Cleaning::Heater	33	6
Tank Cleaning::Pump	21	1
Cargo::Segregations	16	7
Tank Cleaning::Machine	16	4
Inert Gas::Capacity	6	1

The deployed detector configuration is summarized in Table 3.8.

Table 3.8: Training settings for the deployed *LabelFinal* YOLO detector.

Setting	Value
Base architecture	Ultralytics YOLOv8n.
Initial pretrained package	yolo8n.pt
PDF rendering	Render from PDF with target long edge = 3000 px, max long edge = 8000 px, reference PDF long edge = 842.0.
Post-render scale	1.0.
Tile size and overlap	2048 × 2048 px tiles with 512 px overlap.
Epoch budget	100.
Batch size	2.
Page-level filtering	Exclude boxes with more than 50% overlap with prior page-content Table or Text regions.
Image preprocessing	Prior Table / Text areas masked, rotation normalization enabled, and portrait pages optionally padded to a landscape canvas.
Main augmentation settings	Translation = 0.1, scale = 0.5, horizontal flip = 0.5, mosaic = 1.0, close-mosaic after 10 epochs.

Output The information-region detector produced candidate bounding boxes with a target class, confidence score, page association and coordinates. The detected-region and page-summary metadata were added to the page JSONL files. Final Embeddings combined these proposals with text, page-category and page-area information, and the hybrid information-region retrieval stage then scored, filtered and ranked the candidates before excerpt generation.

3.7 Final Embeddings

The outputs from the different models were combined into Final Embeddings, an enriched JSONL metadata file for each page. Figure 3.6 summarizes the page- and text-level fields stored in these files. Page-level records combine categorization, area analysis and detected information regions, while text-level records retain geometry, semantic labels and region scores.

Page	Text
Final Category	Text size
Category area scores	Location on page
Page area scores	Final Category::Subcategory
Page Areas	Area type
Information regions	Region score

Figure 3.6: Page- and text-level information stored in Final Embeddings.

The generated metadata files contain several levels of information produced by the processing pipeline. At token level, each extracted text string is stored together with its location on the page, bounding box coordinates and text-size information. For each token, the final label category is recorded together with category-probability vectors from the three individual experts and a combined probability vector from the fused result. At page level, the final page category is stored together with all related score vectors. The metadata also include page content statistics, information-area statistics and detection-box data from both YOLO models. The detection-box data contain the class, confidence score, coordinates and related metadata for each detected region.

Output The output was one Final Embeddings JSONL metadata file for each page. These files formed the common local input to the information-region retrieval stage. Page categories were used to restrict the search to relevant document types, semantic text labels indicated where particular facts were likely to occur, and spatial information, detected regions and region scores were used to locate and rank candidate excerpts. The selected image excerpts and associated metadata were then used as evidence for the LLM-assisted questionnaire and manual-drafting stages.

3.8 Information Region Retrieval

A hybrid retrieval method was selected because relevant evidence occurs in drawings, tables and ordinary text. It combined visual candidates from the LabelFinal YOLO detector with page- and text-based criteria, including page category, page type and semantic token labels. Class-specific criteria were stored in YAML, which kept the selection logic inspectable and allowed classes to be revised without changing the extraction code.

YOLO boxes provided strong visual evidence, while text-based criteria separated visually similar candidates, for example by checking whether a region referred to a cargo pump. The rules were developed one class at a time with Codex, initially maximizing recall while aiming to keep the FP/TP ratio below 1. Once recall reached approximately 90 %, development focused on reducing false positives without losing coverage. Prioritizing recall reduced the risk of excluding relevant evidence before engineering review.

Output The selected regions were saved in category-specific folders as raster image excerpts, together with an index linking each excerpt to its source page, region and retrieval category. This preserved the visual evidence required by the multimodal LLM. The local evidence-access stage used the index and excerpts to assemble the task-specific evidence supplied to the LLM-generated questionnaire.

Training data for this model were the same labelled information regions described in Section 3.6 and summarized in Table 3.6.

3.9 LLM Agent

For COW manual draft generation, an LLM agent was used to organize evidence, identify project-specific facts and prepare a traceable first draft for engineering review. An LLM was selected for this synthesis stage because instructions can be adapted to heterogeneous document sets and because the model can combine evidence and propose candidate relationships not anticipated by fixed extraction rules. This flexibility is useful when shipyard terminology and available document types vary. It does not transfer engineering authority to the model: the output remains a proposal whose factual basis and derived conclusions must be reviewed.

Fixed rules and templates were preferred for deterministic extraction and formatting wherever they were sufficient. Direct multimodal prompting with complete drawings was excluded by the confidentiality boundary. The smaller GPT-5.4-mini model was explored but not selected for the reported draft because its deductions were inconsistent in the available trial. A locally hosted LLM was considered but not implemented on the CPU-only thesis computer. The reported case-study draft used the configured GPT-5.5 model through the approved cloud interface. These model observations are exploratory and are not a controlled comparison of LLM families. Figure 3.7 expands the LLM-assisted stage and shows the intended flow from locally selected evidence to questionnaire completion, manual drafting and engineering review.

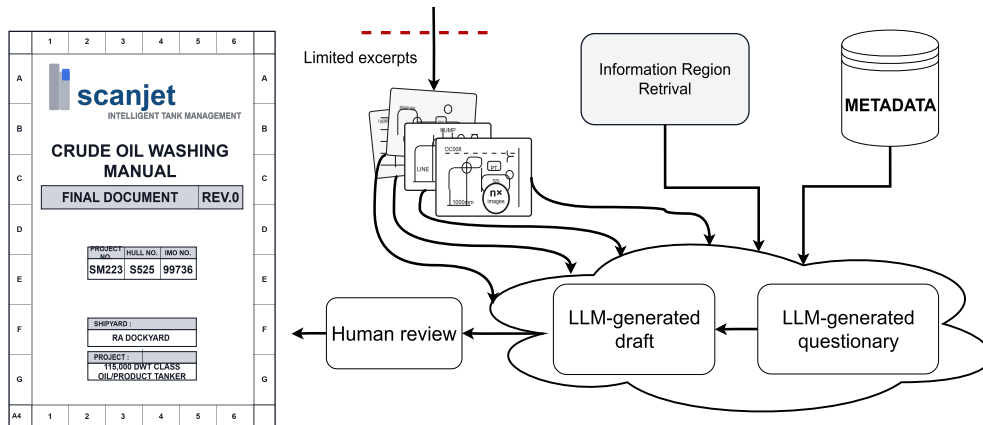


Figure 3.7: Controlled LLM-assisted drafting workflow and evidence boundary.

To support manual drafting, a graphical user interface (GUI) was developed around the LLM agent. The GUI loads and indexes the project folder, indicates which meta-data have been generated, runs the guided-search pipeline, builds a local metadata database, generates cargo-system questionnaires, creates draft manual sections and opens an evidence-based review window. The source material and complete index remain local throughout these operations.

3.9.1 Local Evidence Access and Data Boundary

The LLM was not given direct access to project files. OCR text, page metadata, YOLO-labelled regions, guided-search candidates, crop generation and the searchable metadata database were processed locally. The approved cloud interface received only evidence selected for the current field or drafting task. Table 3.9 defines this boundary.

Table 3.9: Data boundary for the LLM-assisted workflow.

Local only	Eligible for approved upload	Control applied
Complete drawings and original PDF/CAD files	Selected structured facts, such as pump capacities and local equipment properties	Field-specific selection; no unrestricted database access
Full page images and full OCR output	Bounded image excerpts linked to a retrieval result	10 % soft and 20 % hard cumulative exposure limit per source page
Complete spatial layout, CAD layers and piping/system connectivity	Task-relevant evidence assessed as non-reconstructive	Requests exceeding the hard page limit blocked
Completed manuals, questionnaires, Word and Excel files in the thesis experiment	LLM-generated questionnaire fields and their selected evidence	Unresolved fields omitted or flagged for review

The upload assessment was based on reconstructability rather than on whether a value originated in a confidential document. Isolated metadata such as a pump capacity or local component property does not reveal how the complete drawing is laid out or how the piping system is connected. Likewise, a tightly bounded excerpt supplies local visual context without exposing a complete page. The runner tracked the cumulative crop area for each source page: a request crossing 10 % triggered the soft limit, and a request that would take cumulative exposure above 20 % was blocked. This restriction also prevents several overlapping requests from gradually disclosing the entire page. It does not prove that every combination of excerpts is risk-free, so data minimization and approval remain necessary controls.

Completed manuals, completed questionnaires, Word documents, Excel files and original PDFs were excluded from the model input boundary in the thesis experiment. This tested what could be recovered from selected extracted evidence. In a production application, an approved Scanjet questionnaire could be added as a structured cross-check for facts that are difficult to retrieve from drawings, including controls, monitoring, emergency stops and operational constraints.

The agent behaviour was controlled by instruction files in Markdown format. These files defined the evidence rules and output structure. The local folder also contained **Form.csv**, defining the standard variables, and a phrase dataset derived from existing manuals to support consistent wording.

Output The local evidence-access stage produced a task-specific evidence package containing selected structured metadata, source references and bounded raster excerpts. This package was the only project evidence supplied to the approved cloud interface and formed the input to the LLM-generated questionnaire.

3.9.2 LLM-Generated Questionnaire

Before drafting, the system generated an LLM-generated questionnaire that converted selected project evidence into a structured fact sheet. It was organized field by field for facts such as cargo tank count, pump count, capacity, system family, stripping equipment and heating arrangement. Each output was assigned one of three evidence states:

- a value directly supported by one or more identified source locations;
- a relationship or implication proposed from stated premises, requiring engineer confirmation before use; or
- a field for which the available evidence was insufficient or conflicting.

Each field included its proposed value or status, confidence, notes and candidate evidence. This structure allows useful deductions to be surfaced without presenting them as verified source facts. Figure 3.8 shows how supported and unresolved fields were presented in the questionnaire review table.

Status	Group	ID	Value
supported	drawings	drawings.cargo_oil_system	('summary': 'Selected source drawing page for the cargo-sy
supported	tank_arrangement	cargo_tanks.count	('summary': '17 cargo-related compartments identified: 15 o
supported	cargo_pumps	cargo_pumps.count	3 main cargo oil pumps, designated C001F~C003F
supported	cargo_pumps	cargo_pumps.capacity_head	5500 m3/h x 145 mTH per main cargo oil pump
unresolved	cargo_pumps	cargo_pumps.driver_type	
supported	system_family	system.family	Pump-room cargo oil system with shared main cargo oil pu
supported	tank_arrangement	tank.group_division	('summary': 'Cargo tanks/slop tanks are divided into three p
unresolved	tank_arrangement	tank.dirty_slop	
unresolved	tank_arrangement	tank.residual_tank	
unresolved	cargo_pumps	cargo_pumps.controls	
unresolved	cargo_pumps	cargo_pumps.emergency_stops	
unresolved	cargo_pumps	cargo_pumps.monitoring	
unresolved	heating	heating.cargo_tanks	
unresolved	heating	heating.slop_tanks	
supported	stripping_pump	stripping_pump.capacity_head	400 m3/h x 145 mTH
supported	stripping_pump	stripping_pump.driver_type	Steam driven / steam piston type (equipment-list type text n
unresolved	stripping_pump	stripping_pump.controls	

Figure 3.8: Review table for the LLM-generated questionnaire, showing supported and unresolved fields.

Output The output was a structured questionnaire in Markdown format in which each field contained a proposed value or unresolved status, its evidence state, confidence, notes and links to candidate source evidence. The manual-drafting stage used this questionnaire as its fact source rather than reading the complete project documents directly.

3.9.3 Manual Drafting and Review

The drafting agent produced the manual section using the LLM-generated questionnaire as its fact source. Separating evidence interpretation from prose generation made the status of each field inspectable. Unresolved fields were omitted or flagged, and candidate inferences required confirmation before acceptance. The generated draft was presented in a review window so that an engineer could inspect the proposed text and its evidence before including it in a manual. Figure 3.9 shows the draft-review interface, where generated statements are displayed alongside the selected source evidence, source location and corresponding drawing excerpt.

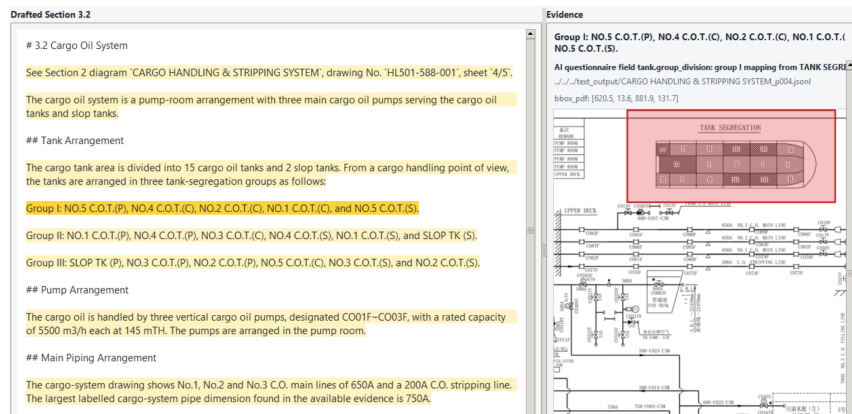


Figure 3.9: Draft-review interface showing generated text and linked source evidence.

Output The output was a proposed draft of Section 3.2, *Cargo Oil System*, in Markdown format. It was presented together with linked source evidence in the review interface, where the engineer could verify, revise or reject the generated statements before any text could be included in the COW manual.

3.9.4 Case-study Evaluation

The LLM-assisted drafting workflow was assessed using one available Chinese-built vessel project with a pump-room-type cargo system, hereafter referred to as the case-study project. The assessment is therefore a proof-of-concept case study, not an independent validation or test, and it does not estimate performance on new projects. No additional projects were available for an independent LLM evaluation within the thesis.

The generated questionnaire and draft were compared by the author with the available Scanjet questionnaire, completed COW manual and source evidence. The comparison considered technical correctness and source support, completeness, conflicting or unresolved fields, traceability and review implications. No separate expert-review dataset or repeated-run experiment was available, so the results are reported as counts and qualitative observations rather than generalizable accuracy percentages.

4

Results

This chapter first presents the documents, data and computational environment used for the analysis, followed by the overall result in relation to the research question. It then presents the detailed results of the implemented pipeline in the same order as the methods chapter. The purpose is to show how each stage contributed to retrieving manual-relevant information and producing the LLM-assisted draft. The chapter ends with the LLM-assisted manual drafting results.

4.1 Documents, Data and Computational Environment

The dataset appendix, Appendix D, lists 32 project entries from Batches 1–4, with 828 pages in total. Three page categories dominated the material: cargo oil system; inert gas and tank ventilation; and water ballast system. Capacity plans were included in 12 of the 32 projects and were comparatively rare at page level. The source projects came from anonymized Chinese and Korean shipyards only. This limited variation is important when interpreting the results: repeated layouts from the same shipyards support evaluation on similar material, but the dataset does not establish generalization to other shipyard standards or document conventions. Table 4.1 summarizes the supplied material and the final processing split.

Table 4.1: Dataset summary from Appendix D and the final processing split.

Item	Value
Appendix project entries	32
Appendix pages	828
Project origin	22 Chinese and 10 Korean entries
Anonymized shipyards	15 identifiers
Existing COW manuals	21 entries
Questionnaires	29 entries
Page categories	17 CAP, 294 COS, 164 WBS, 101 TWS, 171 IGS, 81 UDP
Final processing split	21 training, 5 validation, 10 new/test project roots
Final processing pages	656 training, 67 validation, 289 new pages

All local document processing, model inference, indexing and retrieval were performed on the computer summarized in Table 4.2. The same computer was used for

the processing-time measurements reported later in this chapter. It had no dedicated GPU, so model inference and related processing were performed on the CPU.

Table 4.2: Hardware and software environment used for the reported analysis and computation-time measurements.

Component	Specification
Computer	Dell Pro 13 Plus laptop
Processor	Intel(R) Core(TM) Ultra 5 235U (2.00 GHz)
RAM	32.0 GB (5600 MT/s)
Graphics	Intel(R) Graphics
GPU acceleration	No dedicated GPU available; no GPU acceleration was used
System type	64-bit operating system, x64-based processor
Operating system	Windows 11 Business, Version 25H2

4.2 Overall Result

Overall, the implemented AI-assisted workflow was able to support the production of a COW manual section, but it could not produce a complete and independently usable manual without engineering review. In the one-project case study, the workflow generated a draft of Section 3.2, *Cargo Oil System*, containing the system family, tank arrangement, main cargo oil pump data, major pipe dimensions and stripping-system information.

The page categorization reached 95.5 % accuracy, while the final directed search reached 96.5 % box recall and 94.4 % page recall. The LLM-generated questionnaire contained 24 fields, of which 11 were supported by the available evidence and 13 remained unresolved. Information concerning controls, emergency stops, monitoring, heating and some operational arrangements was therefore omitted or marked for review.

These results show that AI-based techniques can extract and organize relevant information and produce a traceable first draft. However, the workflow should be understood as engineering support rather than autonomous manual generation. The result is based on one drafting case study and does not establish performance on new projects.

4.3 Text Extraction Results

Embedded text was used directly when available, while OCR was required for rasterized content. In the practical tool-selection comparison, the locally containerized Azure OCR implementation recovered 88.0 % of the reference text, compared with 69.0 % for ABBYY and 33.0 % for Tesseract. It was therefore selected for the workflow. This was not a standardized OCR benchmark, and PaddleOCR and EasyOCR were not evaluated. The complete comparison is reported in Table A.1 in Appendix A.

The extraction stage also exposed a limitation relevant to the final workflow: a page could contain mostly embedded text while an important table remained rasterized. Direct PDF extraction alone was therefore not sufficient for every mixed-content page.

4.4 Page-category Results

The final page-category model correctly classified 64 of 67 validation pages, corresponding to 95.5 % accuracy, 96.0 % macro-F1 and 96.5 % balanced accuracy. The remaining errors consisted of two inert-gas pages assigned to the upper-deck category and one cargo-oil-system page assigned to the tank-washing category. Correct categorization reduced the document search space, but an incorrect category could still exclude a relevant page from later retrieval.

The Capacity Plan result was based on only four validation pages and should not be interpreted as stable performance across capacity-plan variants. Detailed model comparisons, the confusion matrix and class-wise metrics are reported in Tables A.2, A.3 and A.4 in Appendix A.

4.5 Word Semantic Labelling Results

The rules, dictionary and LayoutLMv3 branches provided complementary information. Rules were strong for predefined label patterns, the dictionary branch handled close textual matches, and LayoutLMv3 added layout-sensitive information. Their detailed strengths and weaknesses are reported in Tables A.5, A.6 and A.7 in Appendix A.

Table 4.3 shows the overall held-out test result. Fusion reached an F1-score of 79.1 %, compared with 74.5 % for validation-selected best-expert routing. The improvement came mainly from higher precision. Performance still varied substantially between semantic classes; the complete class-wise fusion result is reported in Table A.8 in Appendix A.

Table 4.3: Overall semantic-labelling performance on the held-out test split.

Approach	Recall	Precision	F1
Fusion model	77.9 %	80.3 %	79.1 %
Best-expert routing	77.2 %	72.0 %	74.5 %

4.6 Structural Page Analysis Results

The page-content model classified regions as tables, text, drawings or piping diagrams. Table 4.4 shows that its box-level F1-score was 77.5 %, while the dominant content type was identified correctly for 603 of 620 pages (97.3 %). The dominant-content result was the more relevant measure for later retrieval because it determined how each page was processed.

Table 4.4: Summary metrics for the page-content YOLO detector.

Result	Value
Precision	78.6 %
Recall	76.4 %
F1-score	77.5 %
Dominant content class matched	97.3 %

The remaining dominant-content errors mainly involved drawings confused with tables or piping diagrams. The comparison with the earlier model and the full dominant-content confusion matrix are reported in Tables A.9 and A.10 in Appendix A.

4.7 Target-region Detection Results

The raw LabelFinal YOLO detector supplied candidate regions rather than the final retrieval result. Its performance varied by class: visually distinctive equipment was easier to detect than information presented as ordinary text or tables. Typical errors included missed table-like regions, oversized crops and duplicate detections in dense equipment areas. The combined detector result and class-wise metrics are reported in Tables A.11 and A.12 in Appendix A. These limitations motivated the subsequent area scoring and metadata-guided retrieval stage.

4.8 Area Scoring and Retrieval Results

The final directed-search pipeline combined visual candidates with page categories, semantic text labels and class-specific metadata rules. Table 4.5 shows that it retrieved 96.5 % of the validation information boxes and reached 94.4 % page recall. Candidate precision was 66.5 %, indicating that the method retained most relevant evidence but still returned irrelevant or duplicate candidates for engineering review.

Table 4.5: Final validation summary for actual directed-search candidate classes.

Result	Value
Validation box recall	96.5 %
Validation page recall	94.4 %
Validation candidate precision	66.5 %

Performance differed between information classes, and the two capacity classes required separate proxy-recall measures. The full class-wise retrieval results and capacity proxy results are reported in Tables A.13 and A.14 in Appendix A.

4.9 Pipeline Timing and Practical Performance

Table 4.6 shows that the complete Final Embeddings export required 34.4 seconds per page on the CPU-only laptop. Text processing and LabelFinal YOLO inference accounted for most of this time. The final excerpt export required a further 0.82 seconds per page. These results show that the local pipeline was computationally feasible on the available hardware, although review-time savings were not measured.

Table 4.6: Processing times per page for the main document-analysis stages.

Stage	Time per page
Azure OCR extraction	4.93 s/page
Final Embeddings export (full run)	34.4 s/page
Text stage	21.9 s/page
Page-content YOLO	0.428 s/page
LabelFinal YOLO	12.0 s/page
Finalize/write	0.053 s/page
Final label excerpt export	0.82 s/page

The weighted processing time remained between 33.7 and 39.1 seconds per page across the training, validation and new-project groups. The full project-level timing distribution is reported in Table A.15 in Appendix A.

4.10 LLM-Assisted Manual Drafting Results

The LLM-assisted drafting stage was evaluated as a proof-of-concept case study on one project. The supplied Scanjet questionnaire was excluded from model input and retained for retrospective comparison. The configured GPT-5.5 model used the selected evidence to generate an LLM questionnaire and a draft of Section 3.2, *Cargo Oil System*. The result demonstrates what the workflow produced in this case but cannot establish performance on new projects.

4.10.1 LLM-Generated Questionnaire Results

The generated questionnaire contained 24 fields, of which 11 were supported and 13 remained unresolved. As summarized in Table 4.7, the supported fields covered central system, tank, pump, pipe-dimension and stripping information. Fields requiring control, monitoring, heating or detailed operational information were generally left unresolved rather than filled with unsupported values.

Table 4.7: Assessment of the one-project LLM case study.

Dimension	Case-study observation
Technical correctness and support	The reported tank arrangement, main pump data and several stripping-system facts corresponded to identified source evidence; one questionnaire discrepancy was resolved by checking the base document.
Completeness	11 of 24 questionnaire fields were marked supported and 13 unresolved. Control, monitoring and several operational fields were absent from the draft.
Unsupported or conflicting content	Unresolved fields were generally omitted. The eductor-head discrepancy was retained for source inspection rather than resolved from model confidence.
Traceability	A revision evidence map linked draft statements to questionnaire evidence and bounded local drawing excerpts.
Review effort	A separate timing or expert-rating study was not performed, so no quantitative reduction in engineering review time can be claimed.
External validity	The project overlapped with development material; the case does not measure performance on a new project.

The recovered facts included the pump-room system architecture, 15 cargo oil tanks, 2 slop tanks, three segregation groups, three main cargo oil pumps, major pipe dimensions and stripping equipment. One discrepancy concerned the operating head of the stripping eductors. Inspection of the base document supported the value extracted by the LLM rather than the value in the comparison questionnaire, demonstrating the importance of retaining links to source evidence.

4.10.2 Manual Draft Output

The generated draft captured the high-level structure of the reference section, including tank arrangement, pump arrangement, main piping and stripping. It was technically conservative and generally limited to evidence-supported statements. Control locations, emergency stops, monitoring, heating and several operational arrangements were omitted because they were not supported by the selected evidence. The draft was therefore less complete than a finished COW manual section, but it provided a traceable starting point for engineering review. Because only one project was assessed and review time was not measured, the result supports workflow feasibility but not a claim of general performance or quantified efficiency improvement.

5

Conclusion

The research question was: To what extent can AI-based techniques support the production of COW manuals from CAD-derived drawings and related technical documents with varying formats, layouts and information content? On the evaluated material, the answer is that AI can support important parts of the process by extracting and structuring document content, retrieving relevant evidence and preparing a traceable first draft. The demonstrated workflow is evidence-assisted rather than autonomous. It does not replace engineering interpretation, completeness checks or formal approval, and the one-project drafting case study does not establish performance on new projects.

The strongest quantitative evidence comes from page categorization and directed information retrieval. The page categorizer correctly classified 64 of 67 validation pages (95.5 %), while directed search reached 96.5 % box recall and 94.4 % page recall. Candidate precision was 66.5 %, meaning that the pipeline retained most relevant evidence but still returned irrelevant or duplicate candidates. These results support a reduction in the amount of material that must be searched, but review-time savings were not measured.

The LLM contributed by combining selected facts and proposing relationships that were difficult to represent using fixed rules. In the case study, 11 of 24 questionnaire fields were supported and 13 remained unresolved. This demonstrates the feasibility of evidence-linked drafting, but it also shows that directly extracted facts, inferred relationships requiring confirmation and unresolved fields must remain clearly separated.

5.1 Limitations

The document dataset limits the scope of the quantitative results. It contains 32 listed project entries and 828 pages from anonymized Chinese and Korean shipyards. Repeated structures and terminology are useful for the represented workflow but do not establish generalization to other shipyards, drawing conventions or document qualities. Several categories also had low support; in particular, the Capacity Plan page-category result is based on four validation examples.

The LLM result has a narrower validity boundary. Only the case-study project was available for the reported manual-drafting case study. No additional projects or independent expert-review dataset were available. The case therefore demonstrates that the workflow can produce an evidence-linked draft, but it cannot provide a general accuracy rate, robustness estimate or measured reduction in engineering

review time.

The multi-stage architecture permits error propagation. Missing or incorrect text can affect token labels, page categories, region retrieval, questionnaire fields and the final draft. The workflow also lacks explicit mapping of piping-system connections. This prevents reliable interpretation of several relationships among tanks, pumps, lines and valves and was a major reason that operational fields remained unresolved. Confidentiality imposed a necessary context restriction. Complete drawings, full page images, full OCR output and system-connectivity information remained local, while selected structured metadata and excerpts were supplied to the approved cloud LLM. The excerpts were limited to 10 % soft and 20 % hard cumulative exposure per source page and were assessed as insufficient to reconstruct a drawing or its connection logic. This controlled hybrid design reduces disclosure but also prevents the LLM from seeing some full-system context.

The exploratory comparison between GPT-5.4-mini and the configured GPT-5.5 model was not a controlled model benchmark. The smaller model produced less consistent deductions in the available trial, but this single implementation-specific observation does not show that only larger models can perform the task. Performance may also depend on instructions, retrieved evidence, model version, quantization and domain adaptation. A local model was not evaluated on suitable hardware.

5.2 Future Work

Future work should evaluate models such as ModernBERT and Laya across all text-processing tasks in the project. They are most relevant where TF-IDF is currently used, since their semantic text representations could be compared directly with the existing statistical approach, but they may also be useful for semantic labelling, information retrieval and evidence ranking. The resulting classifiers or retrieval models could also be exposed as tools to the LLM workflow, allowing the LLM to search the project text, rank candidate passages and assess whether retrieved evidence supports a requested manual field. The evaluation should reuse the existing labelled data and consider performance, computational requirements and the feasibility of local deployment. Their suitability should be assessed separately for each task, particularly because text-only models do not represent page layout or graphical relationships (Convai Innovations, 2026; Warner et al., 2025).

The LLM-assisted drafting stage was evaluated on only one project. No newly received project was available for an independent test. Future work should therefore apply the workflow to new projects without adapting it to their content and have the generated drafts independently reviewed for correctness, completeness, traceability and review time.

Explicit connection mapping for P&IDs is another priority. A graph representation built from components, lines, tags and verified relationships could support flow-path and segregation analysis more directly than region retrieval. Because an incorrect connection can have operational consequences, such outputs should expose their source evidence and remain subject to engineering confirmation.

A production workflow should assess the approved Scanjet questionnaire as an additional structured source and consistency check. Deployment alternatives should be

5. Conclusion

evaluated separately: the current approved cloud LLM with minimized evidence, a smaller domain-adapted model and a locally hosted model on appropriate hardware. The comparison should include technical quality, reproducibility, confidentiality, latency, cost and maintenance rather than model size alone.

Bibliography

- AB, S. M.
bibinitperiod S. (2025). Thesis project work agreement.
- AB, S. M.
bibinitperiod S. (2026a, April). *Sc 30t* (tech. rep.).
- AB, S. M.
bibinitperiod S. (2026b). Scanjet - oil & petrochemical.
- Baltrušaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443. <https://doi.org/10.1109/TPAMI.2018.2798607>
- Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C., Ca, J. U., Kandola, J., Hofmann, T., Poggio, T., & Shawe-Taylor, J. (2003). *A neural probabilistic language model* (tech. rep.).
- Bisht, A., Chandrika, K. R., Sharma, D., & Maduskar, D. (2023). Pidpipe: A pipeline for digitizing p
id with unsupervised symbol localization. *2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference, ONCON 2023*. <https://doi.org/10.1109/ONCON60463.2023.10430907>
- Britannica. (2026, April). Tanker - definition, ship, types, & facts. <https://www.britannica.com/technology/tanker>
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4), 18–42. <https://doi.org/10.1109/MSP.2017.2693418>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. <https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html>
- Cao, J., Kwong, S., Wang, R., Li, X., Li, K., & Kong, X. (2015). Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing*, 149, 275–284. <https://doi.org/10.1016/j.neucom.2014.02.072>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., . . . Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*. <https://arxiv.org/abs/2107.03374>

- Convai Innovations. (2026). *Laya: Non-autoregressive decision model* (Version 0.3.20). Retrieved September 26, 2026, from <https://github.com/NandhaKishorM/laya>
- Corel. (2026, April). Autocad drawing database (dwg) and autocad drawing interchange format (dxf) - coreldraw help. <https://product.corel.com/help/CorelDRAW/540111147/index.html?app=CorelDRAW&lang=en#/13TOC495>
- Dietterich, T. G. (2000). *Ensemble methods in machine learning* (tech. rep.). <http://www.cs.orst.edu/%CB%9Ctgd>
- (EIA), U. E. I. A. (2026a, April). Oil and petroleum products explained. <https://www.eia.gov/energyexplained/oil-and-petroleum-products/>
- (EIA), U. E. I. A. (2026b, April). Refining crude oil. <https://www.eia.gov/energyexplained/oil-and-petroleum-products/refining-crude-oil.php>
- Elyan, E., Jamieson, L., & Ali-Gombe, A. (2020). Deep learning for symbols detection and classification in engineering drawings. *Neural Networks*, 91–102.
- Encyclopedia, S. (2026, April). Theories of meaning (stanford encyclopedia of philosophy). https://plato.stanford.edu/entries/meaning/?utm_source=chatgpt.com
- European Committee for Standardization. (2006). Flanges and their joints – pipework components – definition and selection of pn. Retrieved May 12, 2026, from <https://www.sis.se/en/produkter/standardization/vocabularies/fluid-systems-general/ssen13332006/>
- Google Developers. (2026). *Machine learning glossary: Ml fundamentals*. Retrieved May 12, 2026, from <https://developers.google.com/machine-learning/glossary/fundamentals>
- Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022). Layoutlmv3: Pre-training for document ai with unified text and image masking. *MM 2022 - Proceedings of the 30th ACM International Conference on Multimedia*, 4083–4091. <https://doi.org/10.1145/3503161.3548112>
- IBM. (2026, April). What are ai agents? | ibm. https://www.ibm.com/think/topics/ai-agents?utm_source=chatgpt.com
- IMO. (2026a, April). Brief history of imo. <https://www.imo.org/en/about/historyofimo/pages/>
- IMO. (2026b, April). Construction requirements for oil tankers - double hulls. <https://www.imo.org/en/ourwork/environment/pages/constructionrequirements.aspx>
- International Maritime Organization. (1979, November 15). *Revised specifications for the design, operation and control of Crude Oil Washing systems* (Resolution No. A.446(XI)). International Maritime Organization. Retrieved October 4, 2026, from [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.446\(11\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.446(11).pdf)
- International Maritime Organization. (1981, November 19). *Amendments to the revised specifications for the design, operation and control of Crude Oil Washing systems* (Resolution No. A.497(XII)). International Maritime Organization. Retrieved October 4, 2026, from [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.497\(12\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.497(12).pdf)

- International Maritime Organization. (1999, November 25). *Amendments to the revised specifications for the design, operation and control of Crude Oil Washing systems (resolution a.446(xi) as amended by resolution a.497(xii))* (Resolution No. A.897(21)). International Maritime Organization. Retrieved October 4, 2026, from [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.897\(21\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/AssemblyDocuments/A.897(21).pdf)
- International Maritime Organization. (2004, October 15). *Amendments to the annex of the protocol of 1978 relating to the international convention for the prevention of pollution from ships, 1973* (Resolution No. MEPC.117(52)). International Maritime Organization. Retrieved October 4, 2026, from [https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MEPCDocuments/MEPC.117\(52\).pdf](https://wwwcdn.imo.org/localresources/en/KnowledgeCentre/IndexofIMOResolutions/MEPCDocuments/MEPC.117(52).pdf)
- International Maritime Organization. (2026). *Crude oil washing*. Retrieved October 4, 2026, from <https://www.imo.org/en/ourwork/environment/pages/crude-oil-washing.aspx>
- International Organization for Standardization. (1995). *Pipework components – definition and selection of dn (nominal size)* (2nd ed.). Retrieved May 12, 2026, from <https://www.sis.se/en/produkter/fluid-systems-and-components-for-general-use/pipeline-components-and-pipelines/general/iso6708/>
- International Organization for Standardization. (2018). *Iso 21511:2018: Work breakdown structures for project and programme management*. Retrieved September 28, 2026, from <https://www.iso.org/standard/69702.html>
- ISO. (2014). *Diagrams for the chemical and petrochemical industry-part 1: Specification of diagrams*. www.iso.org
- ISO. (2020). *Iso 32000-2:2020. International Organization for Standardization*. <https://www.iso.org/standard/75839.html>
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). *Adaptive mixtures of local experts* (tech. rep.).
- Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 60, 11–21.
- Jurafsky, D., & Martin, J. H. (2026). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models* (3rd ed.) [Online manuscript released January 6, 2026]. Retrieved May 12, 2026, from <https://web.stanford.edu/~jurafsky/slp3/>
- Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998, March). *On combining classifiers* (tech. rep.).
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33. <https://proceedings.neurips.cc/paper/2020/hash/6b493230-Abstract.html>
- Lines, J. (2026, April). Documentation for the json lines text file format. https://jsonlines.org/?utm_source=chatgpt.com
- Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An introduction to information retrieval*.

- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Moreno-García, C. F., Elyan, E., & Jayne, C. (2019). New trends on digitisation of complex engineering drawings. *Neural Computing and Applications*, 31, 1695–1712. <https://doi.org/10.1007/s00521-018-3583-1>
- OpenAI. (2025, October 6). *Codex is now generally available*. Retrieved October 6, 2026, from <https://openai.com/index/codex-now-generally-available/>
- OpenAI. (2026). *ChatGPT capabilities overview*. Retrieved October 6, 2026, from <https://help.openai.com/en/articles/9260256-chatgpt-capabilities-overview>
- Pillai, M. R., Tadepalli, L. N., Bhamre, S. R., Gajbhiye, S. M., & Uplaonkar, D. (2024). Visualizing vitality: Decoding p id connectivity via symbol detection and line analysis. *International Conference on Intelligent Algorithms for Computational Intelligence Systems, IACIS 2024*. <https://doi.org/10.1109/IACIS61494.2024.10722000>
- Powers, D. M. W., & Ailab. (2020). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation.
- Python Software Foundation. (2026). *Python*. Retrieved October 6, 2026, from <https://www.python.org/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*, 779–788. <http://arxiv.org/abs/1506.02640>
- Riley, J. (2017). *Understanding metadata : What is metadata, and what is it for?* NISO Press.
- Salton, G., & Buckley, C. (1987). Term weighting approaches in automatic text retrieval.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1–47. <https://doi.org/10.1145/505282.505283>
- Smith, R. (2007, September). *An overview of the tesseract ocr engine* (tech. rep.). <http://code.google.com/p/tesseract-ocr>.
- Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45, 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- Turpin, M., Michael, J., Perez, E., & Bowman, S. R. (2023). Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-3275>
- Ultralytics. (2025, December). Home - ultralytics yolo docs. <https://docs.ultralytics.com/>
- Urke, T. (1976). Sfi group system—a coding system for ship information [TRID accession number 00148638]. *Preprints of the International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design*. Retrieved September 28, 2026, from <https://trid.trb.org/View/64342>
- Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2023, August). *Attention is all you need* (tech. rep.).

- Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Adams, G. T., Howard, J., & Poli, I. (2025). Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2526–2547. <https://doi.org/10.18653/v1/2025.acl-long.127>
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35. <https://doi.org/10.52202/068431-1800>
- Wikipeda. (2026a, January). .dwg. <https://en.wikipedia.org/wiki/.dwg>
- Wikipeda. (2026b, January). Pdf. <https://en.wikipedia.org/wiki/PDF>
- Wikipedia. (2026a, January). Autocad dxf. https://en.wikipedia.org/wiki/AutoCAD_DXF
- Wikipedia. (2026b, April). Markdown. <https://en.wikipedia.org/wiki/Markdown>
- Wikipedia. (2026c, April). Utf-8 - wikipedia. <https://en.wikipedia.org/wiki/UTF-8>
- YAML Language Development Team. (2021, October 1). *Yaml ain't markup language (yaml) version 1.2.2*. Retrieved May 12, 2026, from <https://yaml.org/spec/1.2.2/>
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., ... Wen, J.-R. (2026). A survey of large language models. <http://arxiv.org/abs/2303.18223>

A

Detailed Validation Results

This appendix contains the detailed model comparisons and class-wise validation results referenced in the results chapter. The main chapter retains the results most directly connected to information retrieval, practical feasibility and the generated COW manual draft.

A.1 OCR Tool-selection Results

Table A.1 reports the complete practical comparison used to select the OCR implementation. The comparison was not a standardized OCR benchmark.

Table A.1: OCR source comparison scores.

Method	Score
Tesseract (free, local)	33.0 %
ABBY (licensed, local)	69.0 %
Azure OCR (local Docker container)	88.0 %
Original (embedded text)	100.0 %

A.2 Page-category Validation

The page-category model variants are compared in Table A.2. The confusion counts and class-wise metrics for the selected model are reported in Tables A.3 and A.4.

Table A.2: Validation accuracy for the page-category model variants.

Model variant	Correct pages	Accuracy	Change
Page TF-IDF only	58/67	86.6 %	Baseline
Page + title hybrid	60/67	89.6 %	+3.0 pp
Three-way mixed before sequence	63/67	94.0 %	+4.5 pp
Three-way mixed + sequence	64/67	95.5 %	+1.5 pp

Table A.3: Page-category confusion matrix for the final mixed three-way model with sequence smoothing. Rows are true classes and columns are predicted classes.

True	CP	COS	IGTV	TWS	UDP	WBS
CP	4	0	0	0	0	0
COS	0	12	0	1	0	0
IGTV	0	0	13	0	2	0
TWS	0	0	0	7	0	0
UDP	0	0	0	0	15	0
WBS	0	0	0	0	0	13

Abbreviations: CP = Capacity Plan; COS = Cargo Oil System; IGTV = Inert Gas and Tank Ventilation Systems; TWS = Tank Washing System; UDP = Upper Deck Plan; WBS = Water Ballast System.

Table A.4: Class-wise validation metrics derived from the final page-category confusion matrix.

Class	Support	Precision	Recall	F1
Capacity Plan	4	100.0 %	100.0 %	100.0 %
Cargo Oil System	13	100.0 %	92.3 %	96.0 %
Inert Gas/Tank Ventilation	15	100.0 %	86.7 %	92.9 %
Tank Washing System	7	87.5 %	100.0 %	93.3 %
Upper Deck Plan	15	88.2 %	100.0 %	93.8 %
Water Ballast System	13	100.0 %	100.0 %	100.0 %
Macro average	67	96.0 %	96.5 %	96.0 %

A.3 Semantic-labelling Validation

Tables A.5, A.6 and A.7 show selected category-level results for the individual experts. Table A.8 reports the detailed class-wise result after fusion.

Table A.5: Held-out test split summary for the rules expert. Categories were ranked by class-wise F1 score.

Group	Label	Supp.	Prec.	Rec.	F1
Top 1	Equipment::Valve	1154	99.5 %	94.4 %	96.9 %
Top 2	Pipe::Dimension	1564	94.5 %	77.3 %	85.0 %
Worst 1	Equipment::Equipment	611	0.0 %	0.0 %	0.0 %
Worst 2	Pipe::Misc	595	0.0 %	0.0 %	0.0 %
Best vs both 1	Equipment::Ballast pump	29	100.0 %	41.4 %	58.5 %
Best vs both 2	Equipment::Cargo stripping	41	69.2 %	43.9 %	53.7 %

Table A.6: Held-out test split summary for the dictionary expert.

Group	Label	Supp.	Prec.	Rec.	F1
Top 1	Pipe::Dimension	1564	73.3 %	95.7 %	83.0 %
Top 2	Equipment::Valve	1154	80.8 %	83.9 %	82.3 %
Worst 1	Equipment::Cargo heating	32	3.9 %	6.3 %	4.8 %
Worst 2	Units::Units	26	3.1 %	73.1 %	6.0 %
Best vs both 1	Units::Mechanics	32	32.3 %	62.5 %	42.6 %
Best vs both 2	Segregation::Segregation	41	72.2 %	95.1 %	82.1 %

Table A.7: Held-out test split summary for the layout expert.

Group	Label	Supp.	Prec.	Rec.	F1
Top 1	Equipment::Valve	1154	89.9 %	95.1 %	92.4 %
Top 2	Pipe::Dimension	1564	92.6 %	90.6 %	91.6 %
Worst 1	Equipment::Cargo heating	32	0.0 %	0.0 %	0.0 %
Worst 2	Units::Mechanics	32	0.0 %	0.0 %	0.0 %
Best vs both 1	Title::Title	87	77.0 %	54.0 %	63.5 %
Best vs both 2	Units::Capacity	91	83.2 %	86.8 %	84.9 %

Table A.8: Class-wise recall, precision, and F1 for the fusion model on the held-out test split, excluding labels with support below 20.

Category	Subcategory	Support	Recall	Precision	F1
Compartments	Ballast tanks	125	73.6 %	78.0 %	75.7 %
Compartments	Cargo tanks	373	83.1 %	84.0 %	83.6 %
Compartments	Compartments	331	68.0 %	76.0 %	71.8 %
Compartments	Fuel tanks	31	51.6 %	94.1 %	66.7 %
Compartments	Misc tanks	21	28.6 %	17.1 %	21.4 %
Compartments	Slop tanks	99	93.9 %	83.0 %	88.2 %
Equipment	Ballast pump	29	51.7 %	35.7 %	42.3 %
Equipment	Cargo heating	32	0.0 %	0.0 %	0.0 %
Equipment	Cargo pump	75	69.3 %	48.1 %	56.8 %
Equipment	Cargo stripping	41	70.7 %	53.7 %	61.1 %
Equipment	Equipment	611	57.3 %	70.0 %	63.0 %
Equipment	Misc pump	37	62.2 %	62.2 %	62.2 %
Equipment	Tank cleaning	82	80.5 %	88.0 %	84.1 %
Equipment	Valve	1154	95.2 %	92.0 %	93.6 %
Medium	Medium	48	52.1 %	43.1 %	47.2 %
Pipe	Dimension	1564	90.3 %	94.6 %	92.4 %
Pipe	Misc	595	66.7 %	65.1 %	65.9 %
Segregation	Segregation	41	39.0 %	100.0 %	56.1 %
Title	Title	87	52.9 %	78.0 %	63.0 %
Units	Capacity	91	86.8 %	84.0 %	85.4 %
Units	Geometric	186	81.2 %	58.5 %	68.0 %
Units	Mechanics	32	0.0 %	0.0 %	0.0 %
Units	Number	62	50.0 %	46.3 %	48.1 %
Units	Units	26	7.7 %	33.3 %	12.5 %

A.4 Structural Page Analysis

Table A.9 compares the earlier and deployed page-content models. Table A.10 gives the dominant-content confusion counts for the deployed model.

Table A.9: Category-wise comparison of the old page-content YOLO model and the latest merged four-class page-content YOLO model on the same 78-image validation split.

Category	GT	Old model			New model		
		P	R	F1	P	R	F1
Table	146	89.3 %	91.1 %	90.2 %	85.6 %	89.7 %	87.6 %
Text	66	67.7 %	31.8 %	43.3 %	79.4 %	81.8 %	80.6 %
Piping diagram	57	67.3 %	64.9 %	66.1 %	80.6 %	87.7 %	84.0 %
Drawing	131	67.4 %	67.9 %	67.7 %	76.8 %	90.8 %	83.2 %

Table A.10: Dominant page-content confusion matrix.

Reference	Table	Text	Piping diagram	Drawing
Table	354	1	1	0
Text	4	21	0	0
Piping diagram	0	0	146	0
Drawing	5	0	6	82

A.5 Target-region Detection

Table A.11 reports the combined raw detector results under the project matching rule. Table A.12 gives the corresponding class-wise detector metrics.

Table A.11: Combined raw YOLO-only performance across label classes. Recall and precision are reported using the project evaluation rule: a prediction is counted as a true positive if the ground-truth box center lies inside the prediction or if IoU ≥ 0.1 .

Result	Recall	Precision
Validation proxy	174/206 (84.5 %)	220/263 (83.7 %)
Total proxy	659/1009 (65.3 %)	824/981 (84.0 %)
Validation actual	174/199 (87.4 %)	220/263 (83.7 %)
Total actual	659/858 (76.8 %)	824/981 (84.0 %)

Table A.12: Per-class validation metrics for the deployed LabelFinal YOLO detector. Classes with zero validation support are excluded.

Category	Subcategory	Support	Precision	Recall	mAP50	mAP50-95
Cargo	Segregations	4	39.0 %	100.0 %	74.5 %	50.0 %
Cargo	Pump	27	63.9 %	39.4 %	48.8 %	22.1 %
Cargo	Stripping	16	75.4 %	76.5 %	83.4 %	33.3 %
Ballast	Pump	16	91.4 %	87.5 %	95.9 %	54.6 %
Ballast	Stripping	22	34.8 %	40.9 %	32.8 %	13.4 %
Tank Cleaning	Heater	4	97.4 %	75.0 %	74.9 %	37.6 %
Inert Gas	PV Valve	15	63.7 %	73.3 %	75.3 %	34.9 %
All	All	104	66.5 %	70.4 %	69.4 %	35.1 %

A.6 Directed-search Class Results

Table A.13 reports the final directed-search result by class. The two capacity classes used proxy-recall statistics and are therefore reported separately in Table A.14.

Table A.13: Per-class validation results for the final directed-search model excluding proxy-only capacity classes.

Category	Subcategory	Box supp.	Validation box recall	Validation candidate precision
Ballast	Pump	17	100.0 %	65.9 %
Ballast	Stripping	7	100.0 %	40.0 %
Cargo	Heating	49	100.0 %	94.6 %
Cargo	Pump	77	94.8 %	80.8 %
Cargo	Segregations	3	100.0 %	100.0 %
Cargo	Stripping	10	100.0 %	32.6 %
Inert Gas	Capacity	1	100.0 %	50.0 %
Inert Gas	PV Valve	21	90.5 %	51.0 %
Tank Cleaning	Heater	6	100.0 %	68.8 %
Tank Cleaning	Machine	4	75.0 %	37.5 %
Tank Cleaning	Pump	4	100.0 %	50.0 %
All	All	199	96.5 %	66.5 %

Table A.14: Separate proxy-recall statistics for the two capacity classes.

Category	Subcategory	Box supp.	Proxy box hits	Proxy box recall
Cargo	Capacity	4	3	75.0 %
Ballast	Capacity	3	2	66.7 %
All	Capacity proxy classes	7	5	71.4 %

A.7 Project-level Processing Times

Table A.15 reports the distribution of final-embeddings processing times across the project groups.

Table A.15: Per-project processing time summary for the full final-embeddings run.

Scope	Projects	Pages	Total (s)			Weighted s/page	Mean s/page
			Mean	Median	SD		
Overall	36	1012	966.5	632.7	1043.4	34.4	36.7
1. Training	21	656	1052.3	602.5	1228.5	33.7	35.5
2. Validation	5	67	523.3	574.0	206.3	39.1	40.5
3. New	10	289	1008.0	818.0	856.2	34.9	37.2

B

Software Inventory for the Metadata Pipeline

This appendix lists all programs developed and used for the whole metadata pipeline in the thesis work as well as their third-party dependencies.

B.1 In-House Programs Used in the Metadata Pipeline

Program	Role in final pipeline	Main package base
TextExtractor	Extracts OCR text and bounding boxes from PDF pages and writes page-level JSONL output. In the final workflow, this stage supplied the base text records used by all later stages.	PyMuPDF, Pillow, Azure AI Document Intelligence, PaddleOCR, EasyOCR, pytesseract
TextLabelerML / TrainFusion3 outputs	Produces word-level semantic labels and probability vectors that are later included in the final-embeddings files and used for final label routing and scoring.	PyTorch, Transformers, scikit-learn, RapidFuzz, sentence-transformers
Page-category model	Predicts page categories using four TF-IDF-based models: page text, title text, PDF name, and page name, plus an ensemble layer. These outputs are included in page metadata and are also used by later capacity-specific rules.	Python standard library, TF-IDF models serialized from earlier scikit-learn-based work

Program	Role in final pipeline	Main package base
Code FinalEmbeddings v3	Builds the integrated page representation used by the final labeling workflow. Each page file includes OCR text rows, text-label outputs, page-category outputs, page-content YOLO outputs, LabelFinal YOLO outputs, and page-level summary statistics.	NumPy, Pillow, PyMuPDF, internal model artifacts
LabelFinal YOLO project runner	Runs the trained LabelFinal YOLO detector on project pages using the high-resolution tiled inference pipeline. Produces box proposals and confidence scores for the direct label classes.	Ultralytics, PyTorch, PyMuPDF, Pillow
TrainLabelFinalAreaScore	Generates candidate label areas and scores them using learned area-level models. Serves as one of the main decision branches in the final hybrid labeler.	PyTorch
TrainLabelFinalMetadataRules	Applies metadata-derived rules with learned weights and a routed hybrid strategy that combines raw YOLO, capacity rules, segregation rules, pump recovery, PV recovery, and area-score outputs. This is the main final label selection layer.	PyTorch, PyMuPDF, Pillow
Retrieval reranker	Reorders candidate excerpts within page/class groups using a learned retrieval-oriented usefulness score. Used after routed prediction to improve the ranking of exported excerpts.	PyTorch
LabelFinal excerpt exporter	Produces the final cropped excerpt images grouped by category under each project folder, together with timing logs and an index file. This is the last operational stage of the pipeline.	PyMuPDF, Pillow, PyTorch, Tkinter for the GUI wrapper

B.2 Third-Party Software used in the Metadata Pipeline

Software package	Use in final pipeline	Reference
Python	Core runtime for all in-house tools.	https://www.python.org/

Software package	Use in final pipeline	Reference
Azure AI Document Intelligence	OCR extraction through the local Azure Document Intelligence container mode.	https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/
Azure Core	Shared SDK infrastructure for Azure AI Document Intelligence.	https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/
PyMuPDF	PDF rendering, page geometry handling, and excerpt image generation. Imported as <code>fitz</code> .	https://pymupdf.readthedocs.io/
Pillow	Image manipulation, masking, crop export, and drawing operations.	https://pillow.readthedocs.io/
NumPy	Numerical operations in embedding export and image-processing support code.	https://numpy.org/
Ultralytics	YOLO-based model inference for page-content and LabelFinal detection.	Ultralytics, 2025
PyTorch	Core deep-learning framework for label scoring, reranking, and model execution.	https://pytorch.org/
scikit-learn	Classical machine-learning support for TF-IDF page categorization and earlier text-label workflows.	https://scikit-learn.org/
joblib	Model serialization and support utilities in the classical ML parts of the workflow.	https://joblib.readthedocs.io/
PaddleOCR	Alternative OCR engine available in the extraction layer.	https://www.paddleocr.ai/
EasyOCR	Alternative OCR engine available in the extraction layer.	https://github.com/JaidedAI/EasyOCR

B. Software Inventory for the Metadata Pipeline

Software package	Use in final pipeline	Reference
pytesseract	Python wrapper for Tesseract-based OCR support in the extraction layer.	Smith, 2007; https://github.com/madmaze/pytesseract
RapidFuzz	String-similarity support in text-label assistance and metadata heuristics.	https://rapidfuzz.github.io/RapidFuzz/
Transformers	Transformer model support in the text-labeling part of the project.	https://huggingface.co/docs/transformers/
sentence-transformers	Text embedding support in the text-labeling and semantic-assistance parts of the workflow.	https://www.sbert.net/
OpenCV	General image-processing support in the YOLO training/inference environment.	https://opencv.org/
SciPy	Scientific computing support in the YOLO environment and auxiliary numeric workflows.	https://scipy.org/
Tkinter	Used only for small GUI wrappers, such as the project excerpt exporter GUI. It is part of the Python distribution rather than a separate pip package.	https://www.python.org/

C

Plotting Drawings with Embedded Text in AutoCAD

This appendix describes a practical procedure used to generate PDF drawings with embedded text from AutoCAD. The purpose of the procedure is to preserve text as machine-readable content in the exported PDF whenever the source drawing and font setup allow it.

C.1 Handling Missing SHX Files

When a drawing is opened and AutoCAD reports missing SHX files, the preferred action is to specify a replacement rather than ignore the warning. Replacing the missing SHX font with a compatible available font increases the chance that text objects are displayed correctly in the drawing and can then be represented properly in the plotted PDF. The replacement-file dialog is shown in Figure C.1.

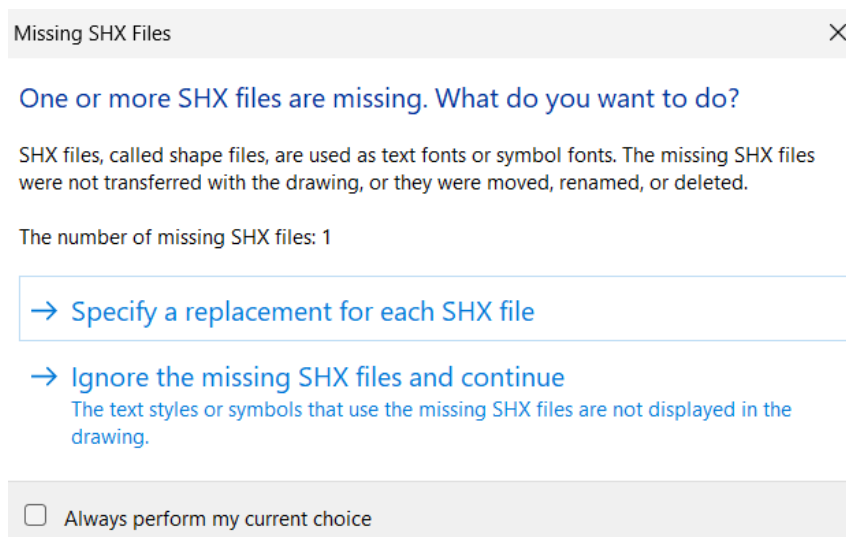


Figure C.1: AutoCAD dialog for selecting a replacement SHX font file.

C.2 PDF Plot Configuration

For drawings intended for downstream text extraction, it is important that the output remains vector-based and that the text is not unnecessarily converted into

image content during export. The AutoCAD PDFSHX setting should be set to 2 so that SHX text is written as searchable PDF text where possible. This can be accessed by just typing PDFSHX in an active drawing window and setting it to 2. After the text resources have been resolved, the drawing can be plotted to PDF. To ensure the correct result, select an AutoCAD PDF plotter, for example AutoCAD PDF (High Quality Print).pc3. The corresponding plot configuration is shown in Figure C.2.

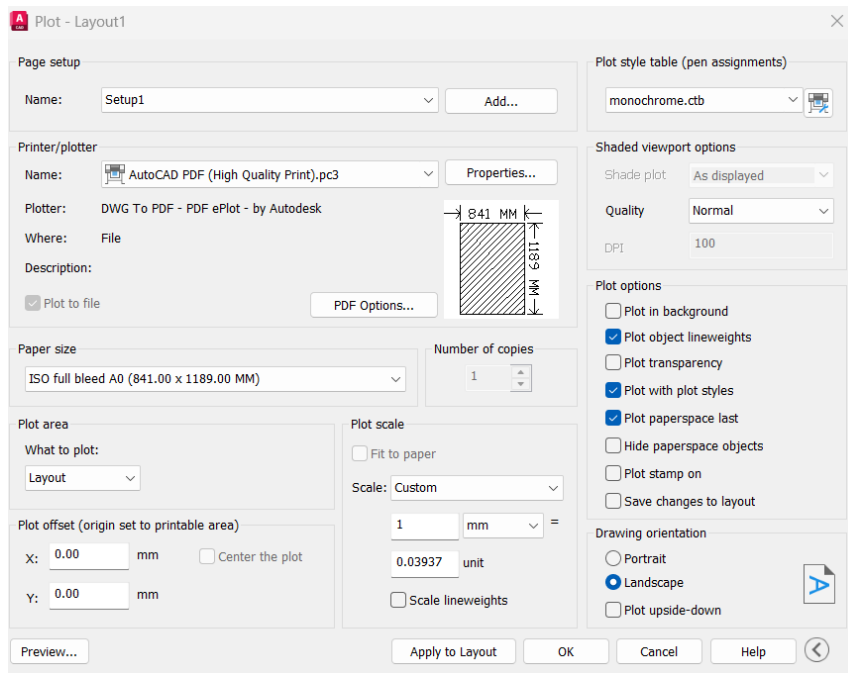


Figure C.2: AutoCAD PDF plot configuration used to preserve machine-readable text.

D

Dataset

The following pages contain the dataset appendix used for this thesis.

Provided dataset

X Code	Project		Embedded text		PDF data types			Pages per category							Documents	
	Batch	Shipyard	Text	CAD	Char	Images	Vectors	CAP	COS	WBS	TCS	IGS	UDP	Total	COW-Manual	Questionary
SM23328	1	Chinese-1			70	0	625004	0	43	21	16	20	1	101	yes	yes
SM24172	1	Korean-1			8805	90	460	0	1	2	1	1	1	6	no	yes
SM24404	1	Korean-1	Yes		25043	0	37124	0	5	2	1	2	1	11	yes	yes
SM24958	1	Chinese-12	Yes		39832	35	77857	0	5	5	3	4	4	21	no	yes
SM00000	1	Chinese-2			0	0	296057	0	2	2	1	2	3	10	no	yes
SM22304	2	Chinese-3			4154	95346	1666480	0	55	15	18	27	0	115	no	yes
SM24195	2	Chinese-13		Yes	14798	0	38664	0	3	1	1	0	1	6	no	yes
SM24238	2	Korean-2			0	57351	58846	1	5	3	2	4	3	18	no	yes
SM24310	2	Korean-3	Yes		37725	0	66139	3	1	3	1	1	1	10	yes	yes
SM24505	2	Chinese-5		Yes	12732	0	212941	0	0	5	0	0	6	11	no	yes
SM24839	2	Chinese-2			0	0	240145	0	6	2	1	3	11	23	no	no
SM25045	2	Korean-1	Yes		41640	0	38470	0	4	3	2	4	0	13	no	yes
SM25099	2	Chinese-6			2128	24	260280	1	5	4	2	3	1	16	no	yes
SM25101	2	Chinese-11			885	225	384700	0	47	25	2	19	1	94	yes	yes
SM25112	2	Chinese-2	Mixed		9135	8	176626	1	2	2	1	3	7	16	no	yes
SM23305	3	Chinese-8			0	8158	75983	0	1	1	1	1	1	5	yes	yes
SM23341	3	Chinese-7		Yes	57966	2	154602	2	5	4	2	2	1	16	yes	yes
SM24088	3	Chinese-1			74	0	80401	0	2	2	1	2	0	7	yes	yes
SM24108	3	Chinese-3			40573	41739	692513	0	23	21	17	21	9	91	yes	yes
SM24115	3	Korean-1	Mixed		19144	13	20721	1	3	2	1	1	1	9	yes	yes
SM24142	3	Chinese-2			8223	0	158862	0	2	2	1	3	7	15	yes	yes
SM24192	3	Chinese-13		Yes	27010	0	72123	0	2	1	1	2	2	8	yes	yes
SM24227	3	Chinese-5			0	0	264196	1	4	2	1	6	2	16	yes	no
SM24310	3	Korean-3	Yes		37725	0	66139	3	1	3	1	1	1	10	yes	yes
SM24786	3	Korean-2	Mixed		59354	8468	74160	1	6	3	2	4	3	19	yes	yes
SM23217	4	Chinese-9		Yes	48977	3	162828	0	7	5	2	6	1	21	yes	yes
SM24272	4	Chinese-9		Yes	0	5	66658	0	1	1	1	4	1	8	yes	no
SM23427	4	Chinese-10	Yes		53957	1	233480	1	5	8	8	6	5	33	yes	yes
SM24147	4	Chinese-11	Yes		81378	0	168789	1	31	3	1	7	1	44	yes	yes
SM24799	4	Korean-1	Yes		68553	0	78120	0	9	6	5	4	2	26	yes	yes
SM25365	4	Korean-2	Mixed		64507	9006	82568	1	6	3	2	4	3	19	yes	yes
SM23391	4	Chinese-8			0	5328	55360	0	2	2	2	4	0	10	yes	yes

CAP - Capacity Plan

COS - Cargo Oil System

WBS - Water Ballast System

TC - Tank Washing System

IGS - Inert Gas and Tank Ventilation Systems

UDP - Upper Deck Plan

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2026

www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY